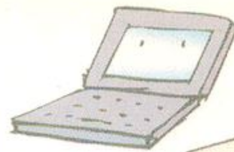
World Wide Web







Hyper Text Markup Language

<HTML>
<HEAD><TITLE>HTML HAND BOOK</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" LINK="#0000FF" VLINK="#551A8B" ALINK="#0000FF">
<META HTTP-EQUIV="REFRESH" CONTENT="5;URL=????.HTML">
</BODY>
</HTML>

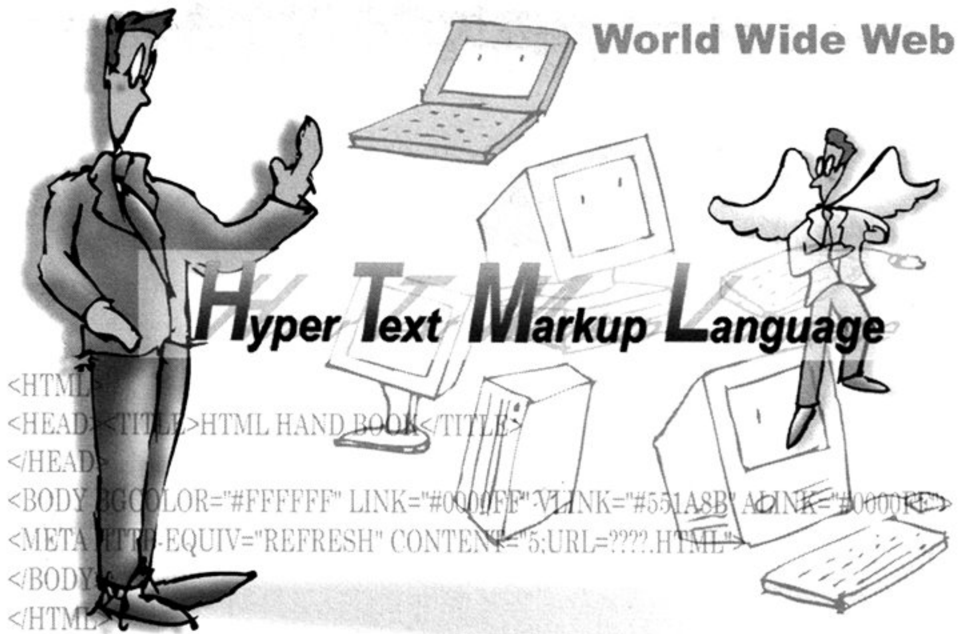
HTML ハンドブック

改訂第2版

渡辺竜生

SOFT BANK Publishing





HTML ハンドブック

改訂第2版

渡辺竜生

はじめに

月日の経つのは早いもので、『HTMLハンドブック』を世に送り出してから、既に1年半が経過しました。その間にインターネットやそれを取り巻く世界も大きく変化し、W3CからはHTML 4.0とHTML文書にレイアウトやアクセシビリティ機能を付加するCSS1・2が正式に勧告されました。

ところが、現在HTML 4.0やCSSに実用的なレベルで対応しているWebブラウザはわずかしかありません。そのため、制作者はあれこれ制限を考えながらHTML 4.0を使用しなければなりません。また、HTML文書の表現力をアップさせるはずのCSSも、Webブラウザ側の不備により、かえってトラブルの種になっていることも事実です（誤解がないように断っておきますが、HTML 4.0やCSSに責任はありません）。

一方、実際にHTML文書を作成してみればおわかり頂けると思いますが、一般的な内容の文書であれば、HTML 3.2で困る場面はそれほど多くはありません。また、近年普及が著しい各種携帯端末に搭載されるWebブラウザの多くは、現在もHTML 3.2を基準に開発されています。であれば、今の段階では「HTML 3.2」というのも、現実的な選択肢の1つだと思われます。その他もいろいろ議論がありましたが、最終的に今回の改訂版は、前版同様HTML 3.2を対象とすることになりました。とはいうものの、多くのWebブラウザがHTML 4.0やCSSに対応するのも、それほど先の話ではないでしょう。ですから、時期が来たら必要に応じてスムーズに移行できるように、HTMLの存在意義や仕様に沿った文書の作成方法を身に付けておくことも大切です。そして、本書がその一助となれば幸いです。

なお、改訂作業に当たっては、前版の読者の方々から頂いたご意見やご指摘を参考に、筆者の誤解やあいまいだった表現を可能な限り改めましたが、まだまだ不充分な部分あるかもしれません。もしお気づきの点がありましたら、ご教示いただけると助かります。

最後になりましたが、本書の制作過程においてご援助を頂いた小沢正直・後藤昌代・嶋岡正二郎の各氏、作例見本に登場して頂いた「water」の皆さん、面倒な編集作業を引き受けてくださったソフトバンク株式会社の浦島弘行氏をはじめ編集部のみなさん、そして前版に対して貴重なご意見を寄せてくださった方々に厚く御礼申し上げます。

1998年9月　渡辺竜生

Contents

HTML HANDBOOK

ビジュアル索引 ―― VIII

第1章　基礎知識 ―― 1

1-1　HTMLについて 2
1-2　HTML文書の要素と属性 4
1-3　HTML文書構造の概略 9
1-4　HTML文書作成上の注意点 12
1-5　Webブラウザ 14
1-6　本書の使い方 15

第2章　HTMLリファレンス ―― 19

基本要素 20

HTML 20
HEAD 21
BODY 22
コメント 29

ヘッダ情報 31

TITLE 31
ISINDEX 33
BASE 35
SCRIPT 38
STYLE 39
META 40
LINK 46

Hn 51
ADDRESS 54

行と段落 56
P 56
BR 60

リスト 63
UL 63
OL 67
LI 71
DL 77
DT 81
DD 82

整形済みテキスト 85
PRE 85

要素のグループ化と位置揃え 88
DIV 88
CENTER 90

引用文 91
BLOCKQUOTE 91

水平線 93
HR 93

表 97
TABLE 97
CAPTION 106
TR 110
TH 113
TD 122

目　次

フォント 131

FONT 131
BASEFONT 135
TT 137
I 139
B 141
U 143
STRIKE 145
BIG 147
SMALL 149
SUB 151
SUP 153

語句への意味づけ 155

EM 155
STRONG 158
DFN 160
CODE 162
SAMP 164
KBD 166
VAR 168
CITE 170

フォーム 172

FORM 172
INPUT 179
SELECT 196
OPTION 200
TEXTAREA 203

リンク 206

A 206

マルチメディア 215

IMG 215

MAP 228
AREA 230
APPLET 236
PARAM 242

拡張 244

拡張要素の注意点 244
SCRIPT 246
NOSCRIPT 250
FRAMESET 252
FRAME 259
NOFRAMES 270
NOBR 273
WBR 275
MULTICOL 277
SPACER 280
MARQUEE 283
BLINK 288
EMBED 290
NOEMBED 297
BGSOUND 299

第3章　HTMLの理解のために ―― 301

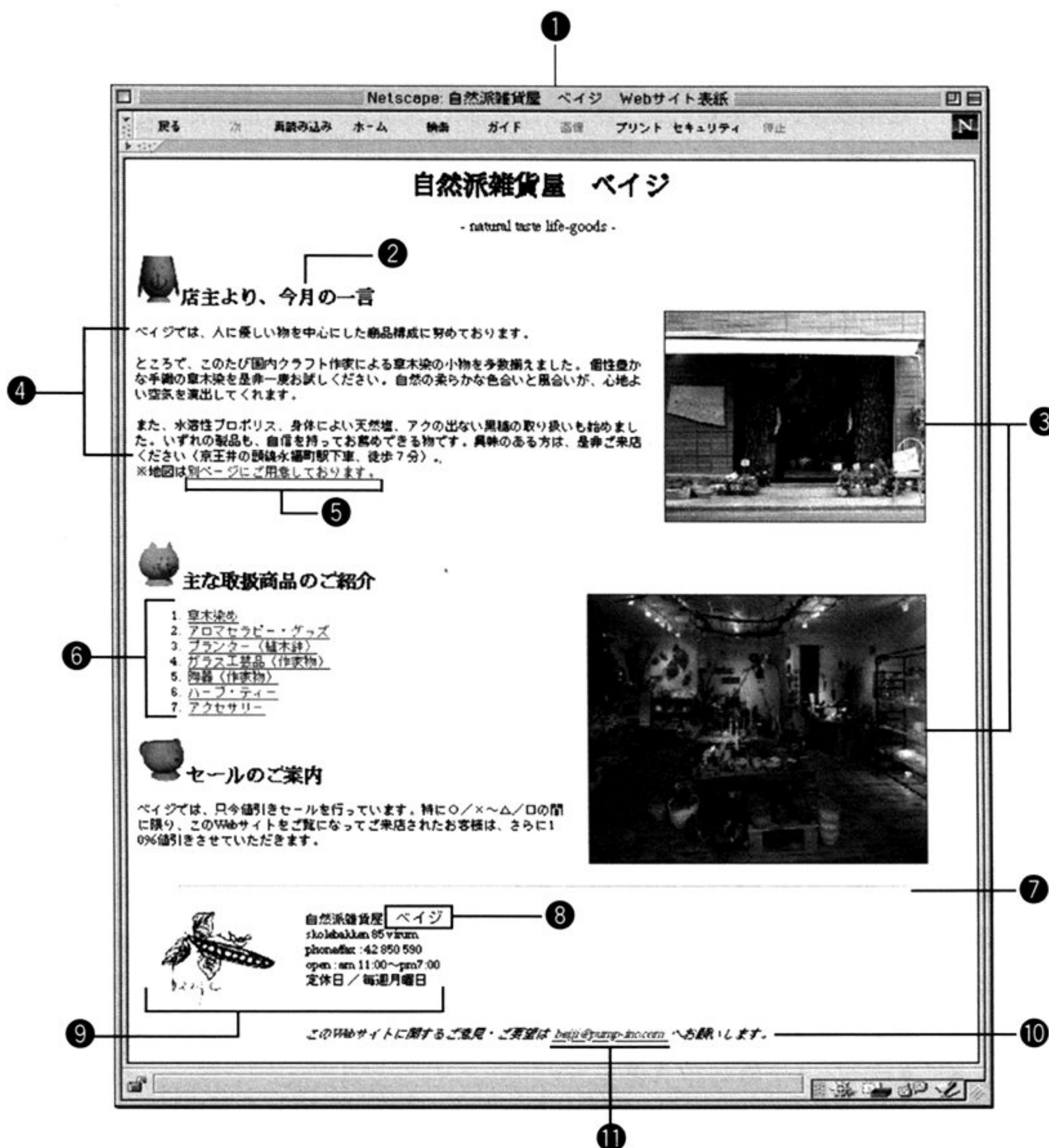
3-1 HTMLの簡単な歴史 302
3-2 SGMLとHTMLの関係 305
3-3 スタイルシートの概要 307
3-4 文字セットと符号化方式 309
3-5 HTMLの文法チェック 312

色名と16進数の対応表 ―― 314

要素引き索引 ―― 316

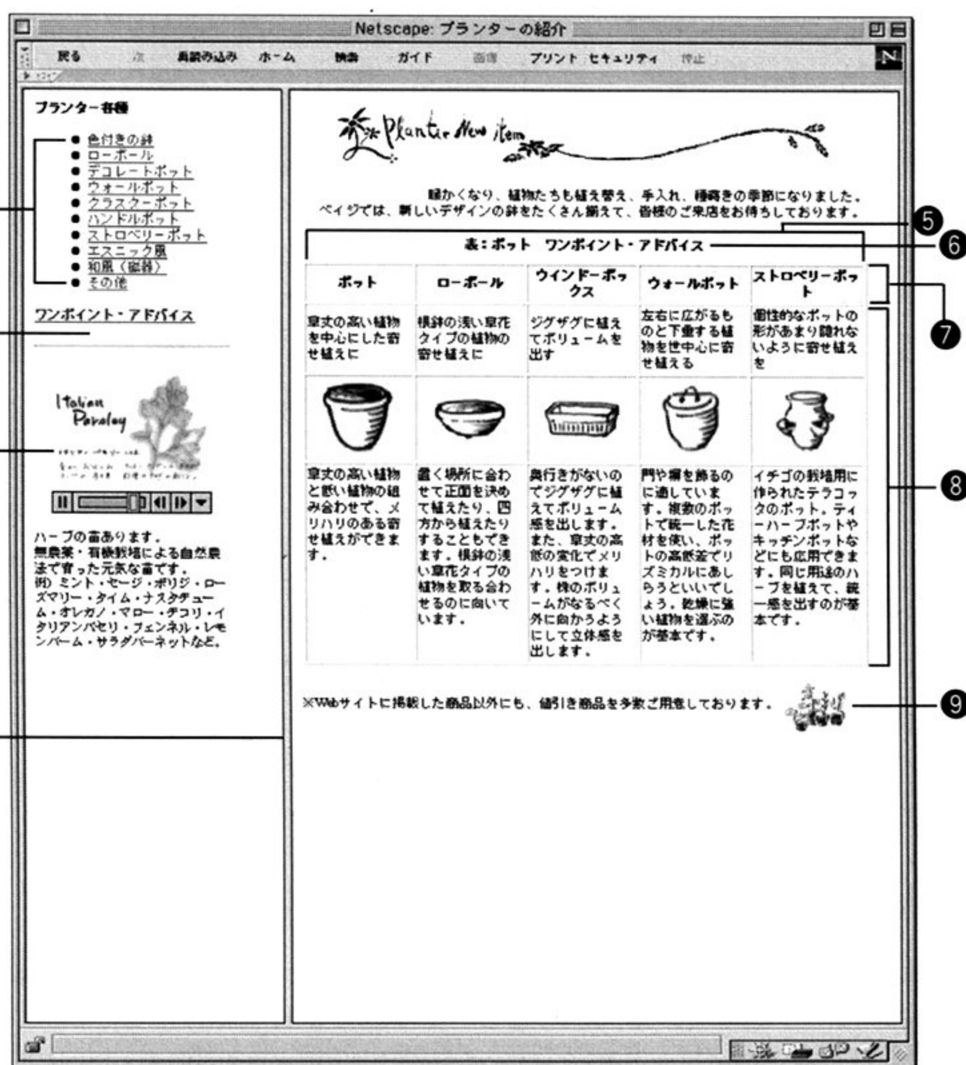
目的引き索引 ―― 321

ビジュアル索引1



❶HTML文書のタイトル……………………P.31
❷見出し………………………………………P.51
❸画像……………………………………P.215
❹段落………………………………………P.56
❺リンク…………………………………P.206
❻順序づけリスト…………………………P.67
❼水平線………………………………………P.93
❽フォントを若干大きく表示…………P.147
❿アドレス…………………………………P.54
⓫メールウインドウを開くリンク……P.209

ビジュアル索引2



❶順不同リスト..............................P.63
❷ターゲットフレームへのリンク..........P.261
❸プラグイン..................................P.290
❹フレーム....................................P.252
❺表組み.......................................P.97
❻表題...P.106
❼表の見出し..................................P.113
❽表のデータ..................................P.122
❾テキスト行内の画像の位置...........P.221

ビジュアル索引3

Netscape: 自然派雑貨屋 ベイジ ーアンケートー

戻る 次 再読み込み ホーム 検索 ガイド 画像 プリント セキュリティ 中止

ご意見&アンケート

このたびは、ベイジのWebサイトをご訪問くださいまして、誠にありがとうございました。
ご意見・ご感想を頂ければ助かります。今後の参考資料とさせて頂きます。
※全項目のご記入は必要ありません。お気軽にどうぞ！

［ご意見欄］
例）もう少し取り扱い商品の幅を広げて欲しい。

［お名前］
［ご住所］
［メール］
［年 齢］
［性 別］ あなたの性別

［興味を持たれた商品］
草木染め
アロマセラピー・グッズ
プランター
ガラス工芸品
陶器

［ご利用の環境］ MacOS UNIX Windows 各種携帯端末 その他

［ご利用のWebブラウザ］ Netscape Navigator Internet Explorer Lynx Cyberdog その他

送る やり直し

beige
自然派雑貨屋 ベイジ

❶ ❷ ❸ ❹ ❺ ❻ ❼ ❽

❶複数行のテキスト入力欄.................P.203
❷１行のテキスト入力欄 P.182
❸選択メニュー................................P.196
❹複数項目の選択ボタン P.183
❺単一項目の選択ボタン P.184
❻フォームの送信ボタン..................P.184
❼入力内容のリセットボタン P.186
❽複数の要素をまとめて位置揃え P.88

第1章

基礎知識



1-1 HTMLについて

WWWとHTML

インターネットを現在のように一般社会に普及させた原動力は、やはりWWW（World Wide Web）の存在でしょう。このWWWの仕組みは、CERN（欧州核物理学研究所、現・欧州共同利用素粒子物理学研究所）のTim Bernes-Lee氏によって考え出されたもので、その主たる目的は「世界中にいる人々が簡単に情報を共有できる」ことにあります。

WWWが登場するまでは、世界中のコンピュータがインターネットで接続されているにも関わらず、情報交換に利用するためにはコンピュータごとに異なる操作方法を習得しなければなりませんでした。しかも、苦労してデータを入手してもそれぞれが独自形式のため、そのまま読めないことが当たり前でした。しかし、WWWのおかげで、現在では世界中のWWWサーバに置かれた情報（リソース）に対して、ユーザの端末や目的の情報の置かれたWWWサーバの機種、場所などは関係なく、簡単にアクセスすることができます。

たとえば、WWWによる情報へのアクセス方法の1つに「ハイパーリンク」があります。このシステムを利用すると、ユーザは画面に表示された「リンク情報」を選択するだけで、目的の情報へアクセスできます。そして、その元になる「ハイパーテキスト」を記述するためのマークアップ言語が「HTML（Hyper Text Markup Language）」で、HTMLによってマークアップされたテキストファイルを「HTML文書」といいます。このHTML文書は単純なテキストデータですから、さまざまな端末で利用することができ、幅広く情報を共有するためには大変都合がいいのです。

本書で扱うHTML

HTMLの公式規格には、現在以下のようなものがあります。

HTML 2.0（1995年11月策定）

現在、ほとんどのWebブラウザやHTML文書を扱うアプリケーションがサポートしており、この規格に沿って作成されたHTML文書の互換性はきわめて高いものがあります。したがって、WWWでの利用には一番適しています。ただ、規格として策定された時期が古いため、いくつかの機能不足な点があり、文書の内容によって

は使用しにくいこともあります。

HTML 3.2（1997年1月策定）

既にWebブラウザメーカーによって開発・普及していた色やフォント、表に関わる部分を容認した規格です。現在ではHTML 2.0に次いで多くのWebブラウザやアプリケーションがサポートしており、文書の互換性の面でも安心して使用できる規格です。ただ、見栄えに関する要素は使い方を誤ると問題を引き起こすので注意が必要です。

HTML 4.0（1997年12月策定）

HTML 3.2を大幅に拡張した規格で、フレームやスクリプトなど、既にWebブラウザメーカーによって開発・普及していた技術を整理・統合して採用したり、別途スタイルシートを併用して、HTML文書に一般文書に近い表現力を持たせることができるようになっています。半面、この規格を完全にサポートするWebブラウザやアプリケーションは現在のところ存在しません。

ところで、本書が対象とするHTMLの規格は最新の4.0ではありません。この点については、企画時に議論があったのですが、結局「従来通りHTML 3.2のまま」となりました。もちろん、HTMLの理想や将来性を考えればHTML 4.0を対象とした方が望ましいでしょう。しかし、現状ではHTML 4.0に「実用的なレベル」で対応している一般ユーザ向けのWebブラウザは、Windows版Internet Explorer 4.0xくらいです。その他のWebブラウザやアプリケーションは、メーカー独自の拡張機能を除けば、依然として対応はHTML 3.2相当といわざるを得ません。それならば、現時点では種々の制限を気にしながらHTML 4.0の文書を作成するよりも、最初からHTML 3.2に従って文書を作成する方が現実的でしょう。もちろん、その一方でHTML 3.2を使用しながらHTMLの意義や基本機能を理解し、将来HTML 4.0へ移行する際のギャップをできるだけ少なくしておくことも大切です。

また、HTML 4.0仕様書の冒頭では、4.0の使用を求めると同時に「過去の文書の互換性を維持するため、HTML 4.0を解釈するツール（Webブラウザなど）は、将来にわたってHTML 3.2と2.0のサポート継続を求めます」とも書かれています。つまり、HTML 2.0と3.2は今後も廃止される予定はなく、作成された文書は将来にわたって有効です。その点はご安心ください。

なお、HTMLの各規格の歴史や位置づけについては、第3章もご覧ください。

1-2 HTML文書の要素と属性

文書と要素

HTML文書の話に入る前に、まず一般的な文書の話をしましょう。短いものですが、以下のサンプル文書を見てください。

```
第１章　QuickTimeの登場
1-1 それは1991年に始まった
今でこそパソコンで動画を再生することは、極めて当たり前のことになったが、
ちょっと前までは本当に難しいことだった。
```

このサンプル文書には「第１章　QuickTimeの登場」という章の見出し、「1-1 それは1991年に始まった」という節の見出し、そして段落が１つあります。これらは、それぞれが文書の中で果たす役割を持った文書を構成する「要素」です（反対に、文書とはこのような「要素」が集まることによってできあがっている、ともいえる）。では、このサンプルをちょっとワープロで修飾してみましょう。

> **第１章　QuickTimeの登場**
>
> **1-1 それは1991年に始まった**
>
> 今でこそパソコンで動画を再生することは、極めて当たり前のことになったが、
> ちょっと前までは本当に難しいことだった。

最初のサンプルよりは、こちらの方が見出しや本文などの要素が修飾（視覚的に区別）されているため、ずっと見やすいと思います。もし、これが印刷するための文書なら、これでめでたしめでたしです。しかし「この文書データを世界中の人と共有しよう」、あるいは「将来使用するコンピュータが変わってもデータを利用したい」となると、話は別です。なぜなら、このようなワープロ独自の修飾を含む文書は、同じワープロソフトでしか再現できないからです。

現在の技術レベルでは、情報を広く共有するためには「テキストデータ」を用いるのが最も確実です。ただ、テキストデータでは文字を修飾したりすることはできません。

要素とタグ

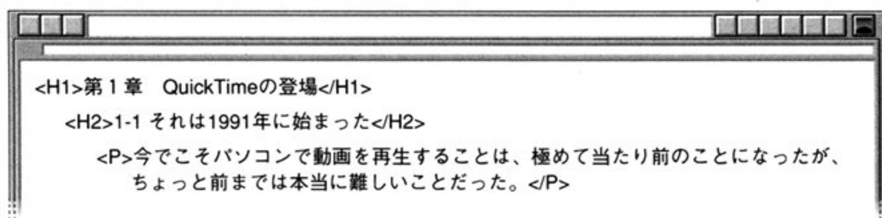
そこで、コンピュータなどがテキストデータ内の各要素や構造を正しく認識できるように、要素の種類を明示する「目印」が考案されました。この目印を「タグ(荷札)」といいます。

では、先のサンプル文書に適当なタグを付けてみましょう(タグは仮に【】で表します)。

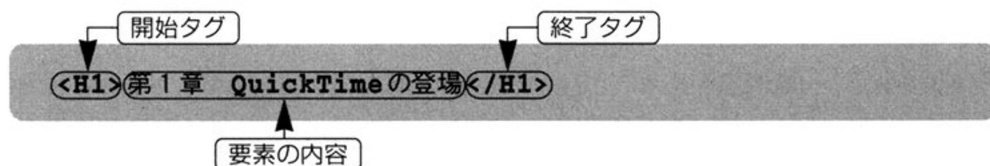
【大見出し開始】第１章　QuickTimeの登場【大見出し終了】
【中見出し開始】1-1 それは1991年に始まった【中見出し終了】
【段落開始】今でこそパソコンで動画を再生することは、極めて当たり前のことになったが、ちょっと前までは本当に難しいことだった。【段落終了】

上図のように、テキストデータに適切なタグを付け、各要素を明示する作業を「マークアップ」といい、そのための言語(命令語)を「マークアップ言語」といいます。そして、HTMLはマークアップ言語の一種なのです。もちろん、タグでマークアップするだけではなく、そのマークアップされたテキストを解釈できるアプリケーションが必要です。そうなってはじめて、コンピュータが文書の要素や構造を自動的に処理できるようになります。

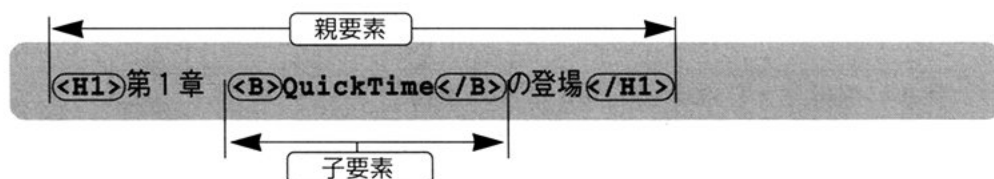
先のサンプル文書をHTMLでマークアップしてみましょう。



何となく雰囲気はおわかりになったと思いますが、１行だけ抜き出してもう少し細かく見てみましょう。



最初の<>で囲まれた部分は「開始タグ」と呼ばれ、要素の始まりを示し、最後の<>で囲まれた部分は「終了タグ」と呼ばれ、要素の終わりを示します。そして、開始タグと終了タグで囲まれた、つまりマークアップされた「第１章　QuickTimeの登場」が、要素の「内容」です。これはテキストだけでなく、他の要素が内容となることもあります。要素の内容に別の要素がなる場合、元の要素を「親要素」、内容となる要素を「子要素」といいます。



ただし、全ての要素を適当に「子要素」として使用することはできません。HTMLの規則では「この要素の内容にできる子要素は○○」と決められています。その大まかな理屈については、P.10を参照してください。また、具体的にどの要素が内容になるのか、どの要素が親要素になるのかについては、２章を参照してください。

なお、要素の種類によっては終了タグが、あるいは開始タグ・終了タグ共に省略可能なものがあります。具体的にどの要素のどのタグが省略可能かについては、同じく2章をご覧ください。

●内容がない要素（空要素）

HTMLの要素の中には内容がないものがあります。それらは「空要素」と呼ばれており、要素の「位置」を表す場合などに使用されます。たとえば、HTML文書はテキストデータですから、画像を中に含むことはできません。そこで、HTML文書内の画像が置かれるべき位置に代わりの空要素を置きます。また、本物の内容との関係については、属性を使用して示します(P.7参照)。

なお、空要素の終了タグはもともと存在しません。具体的にどの要素が空要素であるのかについては、２章をご覧ください。

●内容の形式（空白文字、改行の扱いなど）

要素の内容は、他の要素、あるいは「一般的なテキスト」です。以後、本書の中で使用される「一般的なテキスト」とは、基本的に私たちが日常的に使用する「普通の文字」と思っていただいて結構です。ただし、HTML文書内では普通の文字が「特別な規則」に基づいて解釈されます。

たとえば、空白文字の連続は１個の空白文字として解釈されます。また、タブや

改行コードも空白文字として扱われます。したがって、HTML文書中では適当に改行や空白文字を使用して、メンテナンスしやすいように整形しても構いません。また、開始タグの直後や終了タグ直前の改行コードは無視されます。そのため、開始タグの直後や終了タグ直前で改行して、メンテナンスしやすいように整形することもできます。

ただし、以上の規則は「開始タグの直後や終了タグ直前の空白文字は、改行コードとして解釈され、無視される」ことをも意味します。

```
<P>Sorry, Japanese Only.</P>      1
<P> Sorry, Japanese Only.</P>     2
```

2の「<P>」と「Sorry」の間の空白文字は改行コードとして扱われるため、1と2の表示（解釈）結果は同じになる。

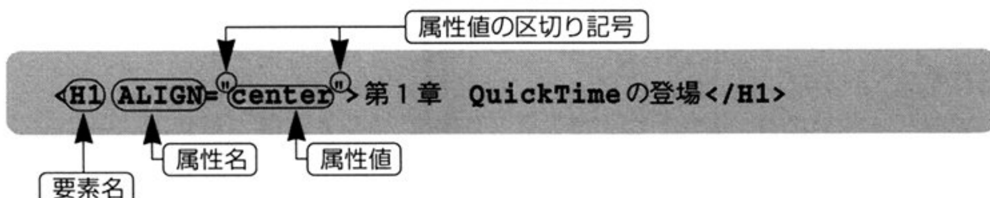
空白文字で単語の区切りを表現しない日本語では、あまり問題が表面化しませんが、欧文をマークアップする場合（特に子要素）は注意が必要です。

さらに、HTMLで特別な意味を持つ記号「<」「>」などをそのまま内容として記述すると、Webブラウザなどは単なる文字ではなく「これはHTMLのタグだな」と解釈します（そうしないと、内容に他の要素を置くことができなくなる）。したがって、これらの特殊記号を表現したい場合は「文字参照」という方法を用います（P.310参照）。

●要素の属性

さて、要素の内容については大体おわかりいただけたと思いますので、今度は要素の「属性」についてお話しします。属性というのは、要素に付加的な情報を与えるための仕組みです。たとえば、属性によって「この要素は（可能ならば）画面の中央に表示しなさい」とか「この要素の元データは○○にある」、あるいは「この要素を表示できないときの代替テキストは○○」などの情報を要素に付加することができます。

属性は要素の開始タグの中、要素名の後ろに記述されます。



前ページの例では、「ALIGN」が属性名で「整列」を意味します。続く「=」は←方向に値を代入することを意味する記号です。そして最後に属性値「center」が置かれます。属性は大文字と小文字の区別はありません。

また、属性値には「"」（二重引用符）あるいは「'」（単引用符）記号で囲まれます。この記号は値がアルファベット（A～Z、a～z）、数字（0～9）、ハイフン（-）、ピリオド（.）の場合に限って省略することが可能です。ただし、いちいち省略可能条件を考えながら文書を作成して結局間違えるより、最初から「全てつけるもの」と覚えた方がはるかに楽で、間違いも起きません。

なお、属性には省略が可能なものが多くあります。具体的にどの属性が省略可能かは、2章をご覧ください。

●属性値の形式

属性値記述には、多くは「一般的なテキスト」が用いられます。空白文字や改行、特殊記号の扱いなどは、要素の内容の「一般的なテキスト」とほぼ同じと考えてください。

ただし、やはり例外はあります。たとえば、META要素の「HTTP-EQUIV属性」「NAME属性」の値の先頭（名前開始文字）は「アルファベット」のみ、続く文字列（名前文字）は「アルファベット、数字、ハイフン、ピリオド」に限定されます。

その他にも、属性によっては大文字と小文字が区別されるものや数字のみが使用されるものもありますが、具体的な例は2章をご覧ください。

●論理型の属性

属性の中には、ある機能のON／OFFを設定する、いわゆる「論理型の属性」があります。論理型の属性は、属性名と同じ名前の値が与えられると「真」、違う値を与えられると「偽」と判断されます。

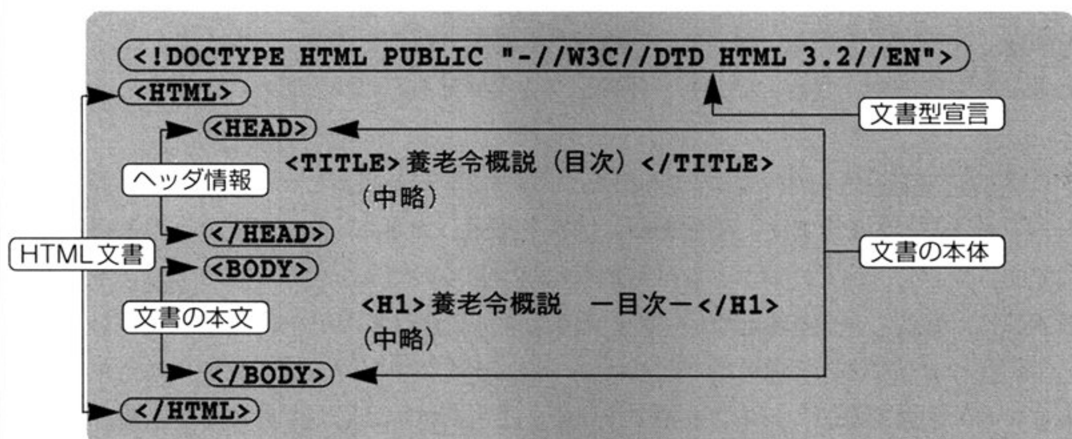
たとえば、INPUT要素には「CHECKED」という論理型の属性があります。この属性を有効にするためには「CHECKED="checked"」と記述するのが正規の書式（full form）です。しかし、現状では正規の書式を理解できないWebブラウザが多く存在するため、一般的には「属性値のみ」の省略書式（minimized form）で記述してください。また、論理型の属性を省略すると、Webブラウザなどアプリケーション側で適当な値が与えられます。そのため「偽」と判断され、属性が無効となります。

なお、具体的にどの属性が論理型になるのかは、2章をご覧ください。

1-3 HTML文書構造の概略

模式図と典型的なHTML文書

HTML文書は、大きく分けて以下のような部分から構成されています。



まず、文書の最初に記述されるのが「文書型宣言」と呼ばれるもので、この文書全体が従う文書型定義（DTD：Document Type Definition　P.305参照）の種類を宣言します。「DOCTYPE」は文書型宣言、「HTML」は文書型定義の名前、「PUBLIC」は公開された規格であることを示します。続く「"-//W3C//DTD HTML 3.2 //EN"」が「W3Cという組織によって策定された文書型定義HTML 3.2の規格」を示します。文書型宣言とは、このHTML文書がどの規格に基づいて作成されたかを明示する、非常に重要な情報ですから、省略することはできません。もちろん宣言を行った以上、文書の内容はその規格に合致していなければなりません。

文書型宣言以降がHTML文書本体、つまり「HTML要素」になります。また、HTML要素は「HEAD要素」と「BODY要素」の2つで構成されています。

HEAD要素はヘッダ情報、つまりタイトルや基準となるURL、文字の符号化方式など文書全体に関わる基本情報です。また、内容は基本的にWebブラウザなどのウインドウには表示されません。

BODY要素は文書本文に相当するもので、通常、Webブラウザなどのウインドウに表示されるのはこの部分です。

ブロックレベル要素とテキストレベル要素

HTML文書の本体であるBODY要素の内容は、機能面から大きく２つのグループに分けられます。

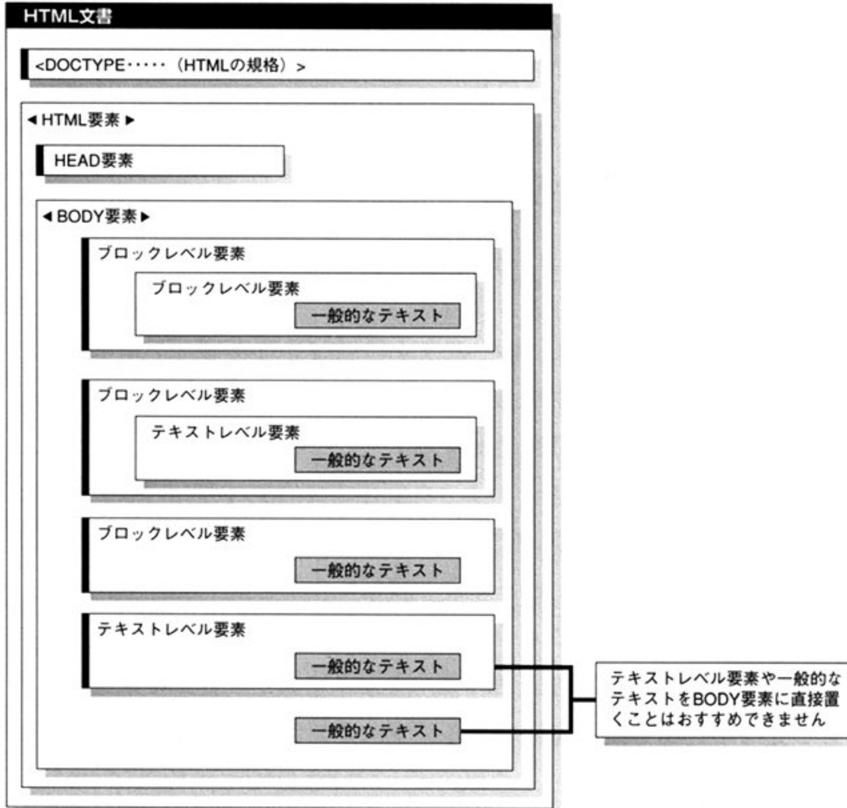
１つは文書本体の意味上の大きな固まりを表す「ブロックレベル要素」で、段落やリスト、引用文、水平線などの要素が該当します。家にたとえると台所や居間、寝室、子供部屋などの区画、あるいはタンスや本棚などの家具に相当します。

もう１つは、文書の具体的なデータにあたる「テキストレベル要素」で、テキストや画像、生のデータあるいはそれらを修飾する要素が該当します。これも家にたとえると本や壁の絵、食器、テレビ、洋服、アクセサリあるいは人間などに相当します。

基本的にブロックレベル要素の中には、ブロックレベル要素とテキストレベル要素を入れることができます。部屋の中には家具や本、おもちゃなどが置けるのと同じです。ただし、同じブロックレベル要素に属するものでも、種類によってはテキストレベル要素しか入れられないものもあります。これは本棚の中に部屋が入らないのと同じようなことと考えてください。一方、テキストレベル要素の中にはテキストレベル要素だけを入れることができます。こちらは本にしおりを挟んだり書き込みをすることはできても、本棚を入れることはできないのと同じようなことと考えてください。

ただ、実際のところどの要素がブロックレベル、あるいはテキストレベルなのか、そして実際に内容として入れられる要素はどれなのか、いちいち覚えておくのは大変かもしれません。そのため、２章では、その要素がブロックレベルとテキストレベルのいずれに属しているのかを記述しておきました。また、具体的に内容として置くことができる要素も全てあげておきました。

なお、HTML 3.2ではBODY要素の中に直接テキストレベル要素を置いても問題ありませんが、HTML 4.0ではブロックレベル要素で構造を組み立てた上で、その中にテキストレベル要素（HTML 4.0では「インライン要素」といいます）を置くように決められています。いわれてみれば、部屋の区画ができあがっていない状態の家にいきなり住む人はあまりいませんから、規則としては、その方が自然でしょう。



ブロックレベル要素とテキストレベル要素の関係例
(実際には内容がない要素もある)

1-4 HTML文書作成上の注意点

最も重要なのは構造を記述すること

前述のように、HTML文書は広く情報を共有するため、あるいは効率よく再利用するためのものです。したがって、HTML文書を作成する場合は「見栄え」ではなく「構造」を記述することを第一番に考えなければなりません。そうはいっても、ピンとこない方も多いかもしれませんが、文書の構造さえはっきりしていれば、たった1つの文書をさまざまな形で利用できるようになるのです。

たとえば、Netscape Navigatorのような画面表示用のブラウザは、「見出し」や「段落」「引用文」などといった要素に合わせて、文字を大きくしたり、揃え位置を変えたりして目立たせ、人間にわかりやすく表示することができます。一方、同じ文書を音声出力用のアプリケーションは、要素に合わせて音を大きくしたり、間を開けたり、適切な説明を挿入して表現することができます。要は中身がしっかりしていればその文書の利用方法は無限である、またさまざまな環境で同じ情報が得られる、ということです。

反対に「見栄え」を中心にHTML文書を作成すると、えてして要素の種類とその内容が一致しない妙なHTML文書ができあがってしまいます。そのような文書は、テキストブラウザや音声出力アプリケーションを使用した場合には、正しく情報が得られなくなってしまいます。また、HTML文書のメンテナンス性も悪くなるので、自分にとってもいいことはありません。

確かに、HTML 3.2には見栄えの制御のための要素や属性が用意されています。ただし、これは「行きがかり上」規格に取り入れられた面もあり、その全ての使用を積極的にすすめることはできません。その使用は、HTML文書の本質を崩さない程度にとどめてください。

アクセシビリティ

前項の内容とも関連しますが、公開するHTML文書の利用者を限定するつもりなのか、それとも幅広く利用してもらうのか、よく考えましょう。公共性の高い官庁や団体ならともかく、個人や私企業が利用者を限定することは、一概に「問題がある」とはいえません。ただし、現在の「インターネット」とは、一般社会に例えれば「公道」に相当するものです。そのような世界に「公開」される文書で、利用者を

限定するのは、やはりいかがなものかと思います。

もちろん、HTMLの規則に完全に従って文書を作成したからといって、全ての人が同じ条件で利用できるわけではありませんが、手始めに以下のような点に気をつけてみてください。

●代替テキストのための属性を積極的に利用する

画像やJava Applet、クライアントサイド・イメージマップ関連の要素には、それらを利用できないWebブラウザのために、代替テキストを記述する「ALT（ALTernate）属性」が用意されています。また、画像の場合は直接Webブラウザで表示できなくとも、受信して別のアプリケーションで表示できることもあるので、ハイパーテキストリンクなどの入手手段も用意しましょう。

●適切なリンクを設定する

リンクを設定する際、「ここ」や「こちら」などといったわかりにくい表現は避け、リンク先の内容が具体的に想像できるものにしましょう。たとえば、音声出力アプリケーションには、リンクが設定されたテキストを抽出して、扱いやすくしてくれるものがあります。しかし抽出したテキストが「ここ」「ここ」「ここ」…では、ユーザは選択不能になってしまいます。

●テキストだけで利用できる方法を用意する

たとえば、画像をリンクのボタンとして使用する機会は多いと思います。その場合は、別途テキストを利用したリンクを用意しましょう。

●横スクロールを避ける

多くのWebブラウザでは、ウインドウの横幅を超える大きさの文書を見るためには、横スクロールが必要となります。これは画面の小さいモニタや端末などを使用している場合、見通しが利かず大変苦痛を感じます。画像などを使用する場合は注意してください。

1-5 Webブラウザ

本書では、現在使用されているWebブラウザのうち、広く一般に普及しているNetscape社の「Netscape Navigator 4.0x」とMicrosoft社の「Internet Explorer 4.0x」を中心にHTML文書の解釈・表示に関して確認作業を行っています。本書のリファレンス中で特に断りなく「Webブラウザ」と表記されている場合は、主にこの2種を指しています。ただし、細かい表示の差に関しては追求していません。なぜなら、「情報の共有」を主目的としたHTMLでは、具体的な表示方法のほとんどは利用する個々のアプリケーションの裁量に任されているからです。

また、本書ではこの種の書籍ではあまり触れられることのない、Lynxなどテキストブラウザでの確認事例も極力載せるように努めました。さらに、わずかではありますが、音声出力を持つテキストブラウザである日本IBM社の「ホームページ・リーダー」での使用結果も内容に反映させました。決して十分とはいえませんが、HTML文書の幅広い利用形態の一部を知っていただければ幸いです。

●本書で確認作業に使用したWebブラウザ

Netscape Navigator 3.0x／4.0x（Mac OS　Windows95）

入手先：**http://home.netscape.com/ja/comprod/mirror/index.html**、他

Microsoft Internet Explorer 3.0x／4.0x（Mac OS　Windows95）

入手先：**http://www.microsoft.com/ie_intl/ja/download/**

Lynx 2.8（MkLinux）

入手先：**http://www.crl.com/~subir/lynx.html**、他

MacLynx 2.7.1b1（Mac OS）

入手先：**http://www.lirmm.fr/~gutkneco/maclynx/**

ホームページ・リーダー（Windows95）

入手先：**http://www.ibm.co.jp/kokoroweb/sns/hpr_down.html**（体験版）

※（）内は使用した環境

1-6 本書の使い方

HTMLの規格について

本書のリファレンス部分は、W3Cによって策定された「HTML 3.2」を対象としています。ただし、HTML 3.2では採用されていないものの、比較的使用頻度が高いスクリプトとフレーム関連の要素については、HTML 4.0に従った簡単な使用方法を紹介しています。なお、最後にWebブラウザメーカーの独自拡張要素を若干取り上げていますが、これはその使用を推奨するものではありません。

見本画面について

見本画面は、特に断りがない限りNetscape Navigator 4.0x (Mac OS版) のものです。また、Mac OS8.1以下では通常日本語フォントのアンダーラインは表示されませんが、本書では別途ユーティリティ (※) を用いて表示しています。

なお、Lynxの見本画面は、Linux上のウインドウシステムのものと、Telnet端末エミュレータを使用したものが混在していますが、実際に動作させているソフトウェア、および表示結果は全く同一のものです。

※江幡 太氏が開発されたフォント変更ユーティリティ「Reliever zeta」
入手先：http://www.eac.co.jp/~ebata/

サンプルソース

サンプルソースの掲載は、スペースや判読性の都合上、解説に必要な部分に限っています。したがって、省略しないことが望まれる要素や属性が省略されている場合もあります。ただし、できるだけ省略していることがわかるように、その旨を注記するようにしました。

用語について

本書は一般向けの書籍という性格上、使用している用語がSGMLやHTML、JISなどの用語、日本語公用文の記法などと一致していない場合もあります。これについてはさまざまな問題があると思いますが、あらかじめご了承ください。

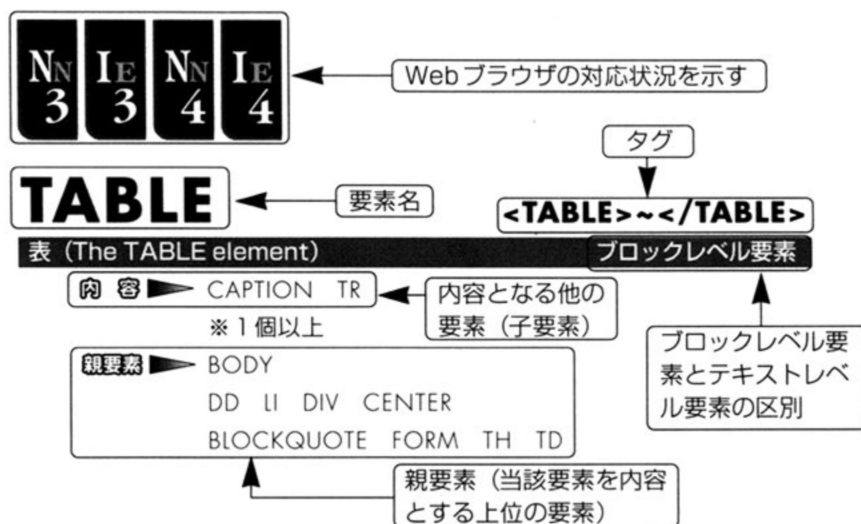
見出しについて

見出しの上部には、Netscape Navigator 3.0xと4.0x、およびInternet Explorer 3.0xと4.0xの対応状況を示すラベルを付けています。

見出しには、要素名とタグの標準的な書式を載せています(終了タグが存在しないものについては、その旨を記述)。また、当該要素の種類(ブロックレベル要素とテキストレベル要素)も示しました(どちらにも属さない要素もある)。なお、要素名は大文字・小文字のどちらを使用しても構いませんが、本書では全て大文字で統一しています。

見出しの下部には、当該要素の内容となる他の要素、及び親要素をHTML 3.2の範囲で列挙しました(※)。列挙した順番は、ほぼHTML 3.2の仕様書に記載されている順序にしたがっています。また、途中の改行は、要素の種類(主にブロックレベル要素とテキストレベル要素)の区切りを表しています。

※DIR・MENU要素は、ほとんど使用されることがないので()付きで列挙しています(P.63参照)。また、XMP・LISTING・PLAINTEXT要素は、新規に作成するHTML文書では使用を避けるよう求められていますので、一覧には載せていません(P.85参照)。



要素の概要

要素の意味や機能の概要を解説しています。また、必要に応じてサンプルソースや見本画面を載せている場合もあります。

要素の概要

TABLE要素はHTML文書で表組を行う要素です。開始タグ<TABLE>と終了タグ</TABLE>によって表の範囲をマークアップします。

属性一覧

要素の属性と簡単な説明、Webブラウザの対応状況を一覧表形式にまとめました。属性名は、大文字・小文字のどちらを使用しても構いませんが、本書では全て大文字で統一しています。また、属性値は、原則として小文字を使用しています(値が固定されていない場合は、適当な変数を値の欄に置き、イタリック表示してあります)。

なお、属性には一般によく知られたものでも、実はWebブラウザメーカーによって独自に拡張されたものもあります。それらを使用すると、HTML 3.2の文書型宣言が行えなくなりますので、属性ごとにHTML 3.2規格に含まれているかどうかも示してあります。

Netscape Navigater 4.0X　Internet Explorer 4.0X

属性名 → **属性一覧**

●ALIGN（初期値：left）　HTML3.2

値	働き	H3	N4	I4
left	左揃え	○	○	○
center	中央揃え	○	○	○
right	右揃え	○	○	○

固定値 → left

属性は省略可

記号について
○：対応
×：未対応（属性がHTML 3.2規格に含まれる場合）
－：未対応（属性がHTML 3.2規格に含まれない場合）
！：Webブラウザメーカーが独自に拡張した属性

属性解説

属性一覧で挙げた属性について、その意味や具体的な使用方法・結果を解説しています。また、必要に応じて、サンプルソースや見本画面を載せています。

属性解説

●ALIGN属性

表の位置揃えを指定します

ALIGN属性を省略した場合、Webブラウザなどアプリケーション側でleftが与えられます。

```
<TABLE ALIGN="right">  ← ALIGN属性の指定
      <TR><TH> A </TH><TH> B </TH><TH> C </TH><TH> D
</TH></TR>
      <TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD><TD> 4
</TD></TR>
      <TR><TD> 5 </TD><TD> 6 </TD><TD> 7 </TD><TD> 8
</TD></TR>
</TABLE>
```

重要な用語

・IETF（Internet Engineering Task Force）

インターネット技術特別調査委員会。インターネット上で使用されるプロトコルなどさまざまな規格案を出して標準化を行い、それをRFC（Request for Comments）として公開しています。

・W3C（World Wide Web Consortium）

WWWの標準化と推進のために設けられた国際的な学術研究組織で、現在では200を超える企業や団体、政府機関などが会員として参加しています。また、ホスト機関としては、マサチューセッツ工科大学計算機科学研究所（MIT/LCS）、フランス国立情報処理自動化研究所（INRIA）、慶應義塾大学SFC研究所の3つがあり、それぞれの担当地区（北米、欧州、アジア）のW3Cの活動をとりまとめています。

・ISO（International Standards Organization）

国際標準化機構。1947年に設立された、工業製品の国際的な統一規格を策定に関わる機関。また、1987年から電気・電子部門はIEC（International Electrotechnical Commission：国際電気標準会議）が担当しています。

・プロトコル（Protocol）

ネットワークは、単に物理的にケーブルを接続しただけでは使用できません。実際にデータのやり取りを行うためには、そのための手順などがきちんと規格化されていなくてはなりません。この規格化された手順をプロトコルといいます。

・HTTP（Hyper Text Transfer Protocol）

HTML文書の転送に使用されるプロトコル。

・Webブラウザ（WWW Browser）

HTML文書を読み込んで、マークアップを手がかりに文書構造を解釈して利用するアプリケーションのうち、モニタ画面に表示することを主目的としたものを一般に「Webブラウザ」と呼んでいます。

・リソース（Resource）

WWWでは情報資源のことを指します。HTML文書はもちろん、画像や動画、サウンド、Java Appletなどさまざまなデータがリソースとなります。

・URL（Uniform Resource Locator）

WWWサーバにあるリソースのありかを示すための文字列。URLの先頭はリソースへのアクセスに使用するプロトコル名（スキーム：Scheme）とコロン（:）で始まります。プロトコルはHTTP以外にもFTPやSMTP、TelnetなどWWWで使用可能なものであれば全て記述できます。つまりプロトコルが異なっても、リソースへのアクセス方法は統一的に表現可能です。最近のWebブラウザはさまざまなプロトコルに対応しているため、そのありがたさをより実感できるでしょう。

第2章

HTMLリファレンス



NN 3 IE 3 NN 4 IE 4

HTML

<HTML>~</HTML>

HTML（The HTML element）

内容 ▶ HEAD BODY

親要素 ▶

要素の概要

HTML要素は、文書型宣言以降のHTML文書本体です。開始タグ<HTML>と終了タグ</HTML>の間には、HTML文書のヘッダ情報であるHEAD要素と、HTML文書の本文であるBODY要素が置かれます。

開始タグ<HTML>と終了タグ</HTML>は共に省略可能です。これはHTML文書の構造上その存在が推測可能なためですが、ソースの見やすさを考慮して省略しないようにしましょう。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
  <HTML>  ←── 開始タグ
      <HEAD>
          HEAD要素（HTML文書のヘッダ情報）
      </HEAD>                                  ↑
      <BODY>                               HTML要素
          BODY要素（HTML文書の本文）             ↓
      </BODY>
  </HTML> ←── 終了タグ
```

属性一覧

なし

属性解説

なし

HEAD

<HEAD>~</HEAD>

ヘッダ（The HEAD element）

内容 ▶ TITLE ISINDEX BASE STYLE SCRIPT META LINK

親要素 ▶ HTML

要素の概要

HEAD要素は、HTML文書のヘッダ情報です。

開始タグ<HEAD>と終了タグ</HEAD>の間には、TITLE要素やBASE要素などHTML文書全体に関する基本情報を記述します。その基本情報はBODY要素全体に適用されます。

なお、開始タグと終了タグは共に省略可能です。これはHTML文書の構造上その存在が推測可能なためですが、ソースの見やすさを考慮して省略しないようにしましょう。

```
<HEAD>  ◀── 開始タグ
    <TITLE>養老令概説</TITLE>
    その他のHEAD要素
</HEAD>  ◀── 終了タグ
```

属性一覧

なし

属性解説

なし

N3 IE3 N4 IE4

BODY

<BODY>~</BODY>

ボディ (The BODY element)

内容 ► H1　H2　H3　H4　H5　H6
ADDRESS
P　UL　OL　DL　(DIR)　(MENU)　PRE　DIV
CENTER　BLOCKQUOTE　FORM　ISINDEX　HR
TABLE
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
APPLET　IMG　FONT　BASEFONT　BR　MAP　INPUT
SELECT　TEXTAREA　SCRIPT
一般的なテキスト

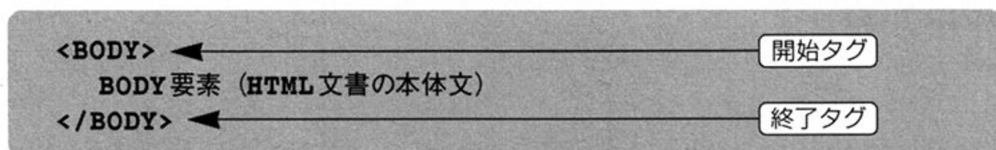
親要素 ► HTML

要素の概要

BODY要素は、HTML文書の本文です。

開始タグ<BODY>と終了タグ</BODY>の間には、HTML文書の本文を記述します。

開始タグと終了タグは共に省略可能です。これはHTML文書の構造上その存在が推測可能なためですが、ソースの見やすさを考慮して省略しないことが普通です。



注意 HTML 3.2では、BODY要素の中に直接テキストレベル要素を置いても問題はありません (P.10参照)。ただし、文書の構造としてはいったんブロックレベル要素で区画を作り、その中に具体的なデータであるテキストレベル要素を置く方が自然です。

属性一覧

●BGCOLOR

値	働き	H3	N4	I4
color	背景色を指定	○	○	○

属性は省略可

●TEXT

値	働き	H3	N4	I4
color	テキスト色を指定	○	○	○

属性は省略可

●LINK

値	働き	H3	N4	I4
color	未アクセスのリンク色を指定	○	○	○

属性は省略可

●VLINK

値	働き	H3	N4	I4
color	既アクセスのリンク色を指定	○	○	○

属性は省略可

●ALINK

値	働き	H3	N4	I4
color	アクセス中のリンク色を指定	○	○	○

属性は省略可

●BACKGROUND

値	働き	H3	N4	I4
url	背景イメージに使用する画像のURLを指定	○	○	○

属性は省略可

●BGPROPERTIES

値	働き	H3	N4	I4
fixed	背景イメージを固定	–	–	!

属性は省略可

●TOPMARGIN

値	働き	H3	N4	I4
pixels	上側マージン	−	−	!

属性は省略可

●LEFTMARGIN

値	働き	H3	N4	I4
pixels	左側マージン	−	−	!

属性は省略可



color=色指定の16進数（#rrggbb）、または色名　*url*=URL
pixels=pixel数

属性解説

●色の指定方法について

基本的に先頭に「#」を付加した数値で指定します。その数値はRGB（光の３原色）の各色を256段階にし、16進数（0〜9の数字とA〜Fのローマ字）２つの組み合わせ（16×16=256）で表現したもので、理論上256×256×256=16,777,216色を表現することができます。しかし、現実的にはHTML文書を表示する端末（コンピュータとは限りません）が16,777,216色の表示が可能とは限りませんし、たとえ可能であっても正確に再現することはできません。そのため、あまり凝った色は使用しない方が無難です。

なお、HTML 3.2では、基本的な16色（WindowsのVGAパレットに採用されている標準色）に関しては、16進数だけでなく、色名による指定方法も可能です。

	16進数指定	色名
黒色	COLOR="#000000"	COLOR="black"
銀色	COLOR="#C0C0C0"	COLOR="silver"
灰色	COLOR="#808080"	COLOR="gray"
白色	COLOR="#FFFFFF"	COLOR="white"
栗色	COLOR="#800000"	COLOR="maroon"
赤色	COLOR="#FF0000"	COLOR="red"
紫色	COLOR="#800080"	COLOR="purple"
紫紅色	COLOR="#FF00FF"	COLOR="fuchsia"
緑色	COLOR="#008000"	COLOR="green"
ライム色	COLOR="#00FF00"	COLOR="lime"
オリーブ色	COLOR="#808000"	COLOR="olive"
黄色	COLOR="#FFFF00"	COLOR="yellow"
濃紺色	COLOR="#000080"	COLOR="navy"

	16進数指定	色名
青色	COLOR="#0000FF"	COLOR="blue"
暗緑色	COLOR="#008080"	COLOR="teal"
水色	COLOR="#00FFFF"	COLOR="aqua"

その他の色に関しても色名で指定できる場合がありますが、Webブラウザメーカーによる独自拡張ですから、再現性の保証はありません。通常は16進数で指定してください(独自拡張の色名と16進数の対応表はP.314参照)。

●BGCOLOR属性

Webブラウザで HTML文書を表示する場合の背景色を指定します。色は16進数、あるいは色名で指定します。

●TEXT属性

Webブラウザで HTML文書を表示する場合のテキスト色を指定します。BGCOLOR属性やBACKGROUND属性で背景色を変更し、テキストが読みにくくなる場合などに使用します。

色は16進数、あるいは色名で指定します。

●LINK属性

Webブラウザで HTML文書を表示する場合、まだアクセスしていないリンクのテキストやボーダー(画像の枠など)色(ホットテキスト色)を指定します。

色は16進数、あるいは色名で指定します。

●VLINK属性

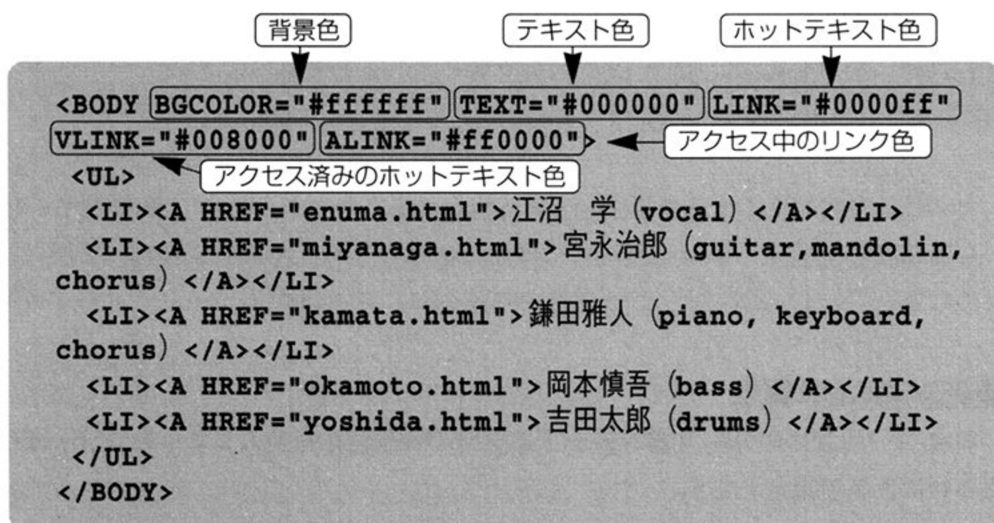
Webブラウザで HTML文書を表示する場合、既にアクセスしたことがあるリンクのテキストやボーダー(画像の枠など)色(ホットテキスト色)を指定します。

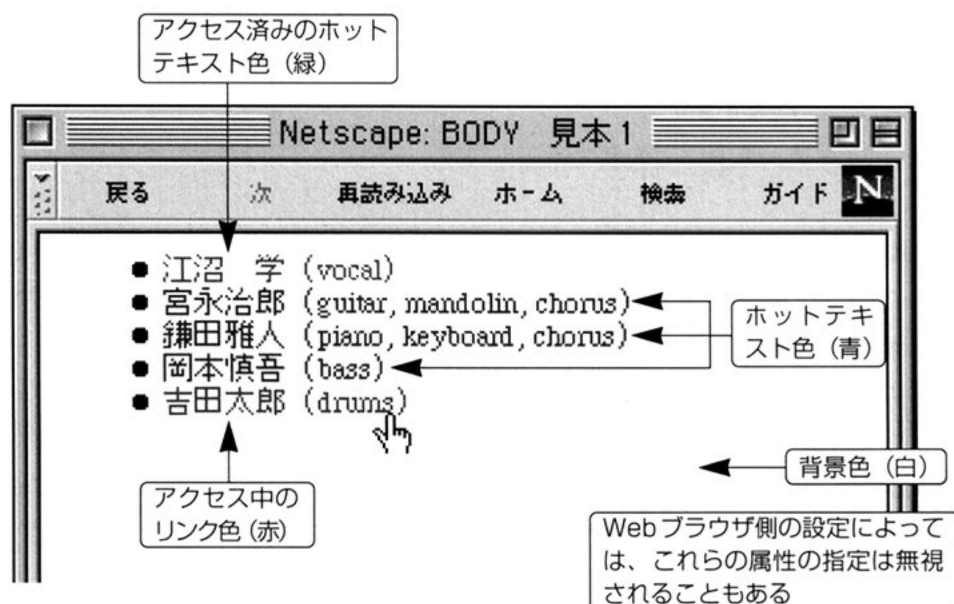
色は16進数、あるいは色名で指定します。

●ALINK属性

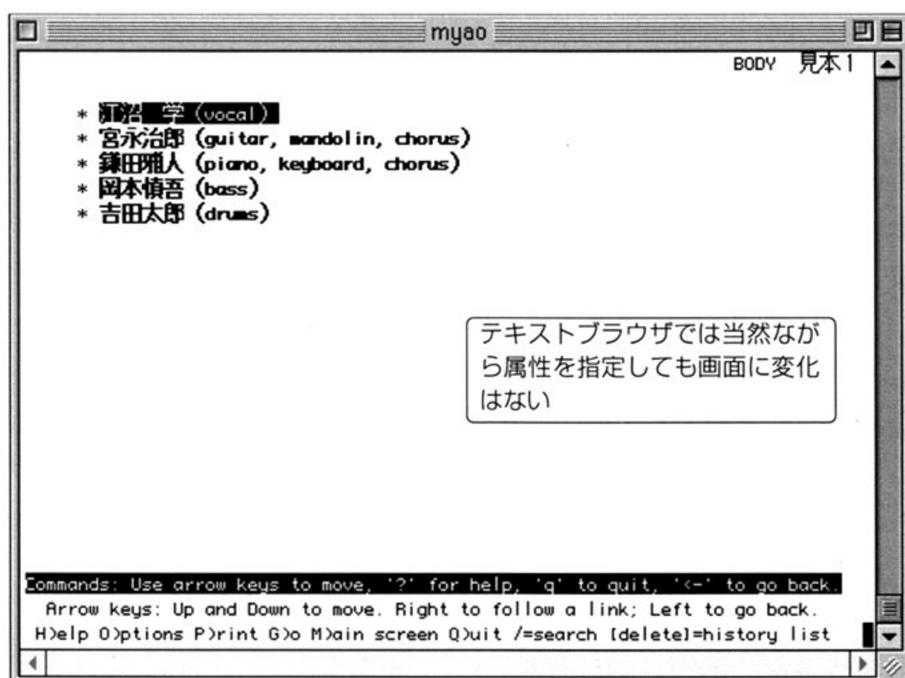
Webブラウザで HTML文書を表示する場合、アクセス中(マウスなどでクリックしたとき)のリンクのテキストやボーダー(画像の枠など)色(ホットテキスト色)を指定します。

色は16進数、あるいは色名で指定します。





各種オブジェクトの表示例



Lynxでの表示例

●BACKGROUND属性

Webブラウザで HTML文書を表示する場合、背景にGIFやJPEG形式の画像を使用することができます。その際、BACKGROUND属性で画像のURLを指定します。

BACKGROUND属性の指定

```
<BODY BACKGROUND="logo1.gif">
 <UL>
  <LI><A HREF="enuma.html">江沼　学（vocal）</A></LI>
  <LI><A HREF="miyanaga.html">宮永治郎（guitar, mandolin,
chorus）</A></LI>
  <LI><A HREF="kamata.html">鎌田雅人（piano, keyboard,
chorus）</A></LI>
  <LI><A HREF="okamoto.html">岡本慎吾（bass）</A></LI>
  <LI><A HREF="yoshida.html">吉田太郎（drums）</A></LI>
 </UL>
</BODY>
```



イメージを利用した背景

●BGPROPERTIES属性

背景イメージがスクロールしないように固定します。テキストや通常の画像などは固定されずスクロールします。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●TOPMARGIN属性

上側マージンをpixel数で指定します。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●LEFTMARGIN属性

左側マージンをpixel数で指定します。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

コメント

<!--~-->

コメント (Comments)

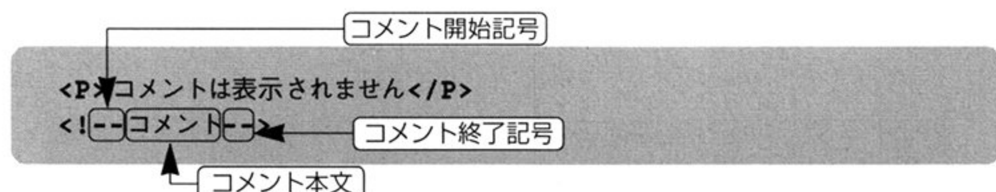
内容 ▶ 一般的なテキスト

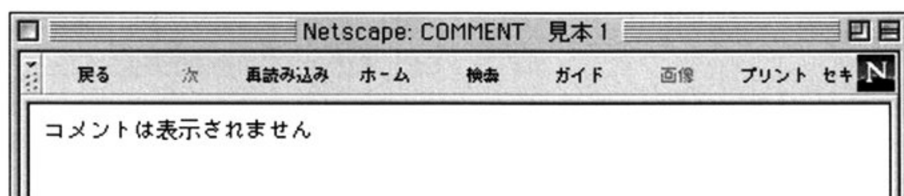
要素の概要

HTML文書内にコメントを記述する場合に使用する記号です。

「<!--」と「-->」に囲まれた部分は、Webブラウザではコメントとして解釈され表示されません。

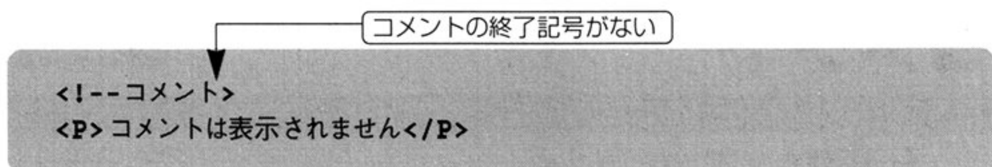
また、スクリプト言語やスタイルシートなどは、それらに未対応のWebブラウザで不用意に表示されるのを避けるため、コメントとして記述されます(P.247参照)。



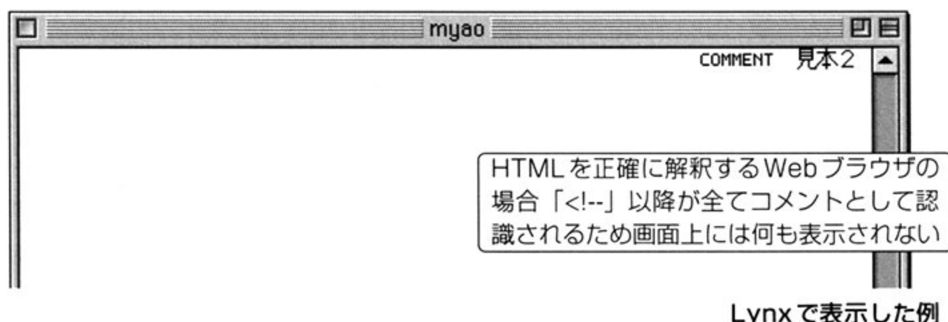


コメントは表示されない

「<!」はマークアップ開始記号、「--」がコメントの開始記号、そしてコメント本文、コメントの終了記号「--」が続き、最後にマークアップ終了記号である「>」が置かれます。つまり「--」~「--」で1組のコメントとして扱われます。もし、コメントの終了記号「--」が記述されていないと、それ以降に続くテキストは全てコメントとして扱われてしまいます。コメントを記述する際には、必ず「--」~「--」がセットになっているかどうか気をつけてください。



間違ったコメントの記述例

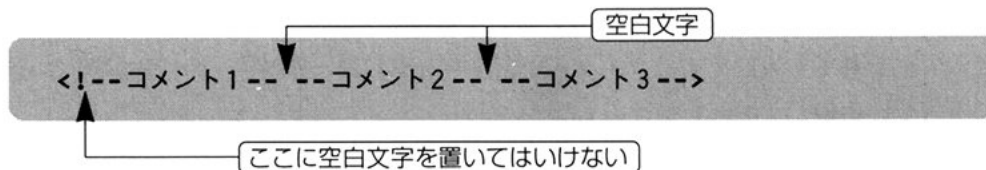


Lynxで表示した例



Netscape Navigator4で表示した例

また、コメントの後ろ側（コメントの終了記号「--」の後ろ）には空白文字を置くことができます（空白文字以外は不可）。そのため、複数のコメントを並べて記述することもできます。ただし、コメントの前側（「<!」と「--」の間）には、空白文字も置くことはできません。



TITLE

<TITLE>~</TITLE>

タイトル（The TITLE element）

内 容 ▶ 一般的なテキスト

親要素 ▶ HEAD

要素の概要

TITLE要素は、全てのHTML文書に必須のヘッダ情報です。

開始タグ<TITLE>と終了タグ</TITLE>によって、タイトルをマークアップします。TITLE要素の内容には、一般的なテキストが使用可能です（ただし、スクリプト言語の類は記述できない）。また、その性格上、HTML文書内に1つしか置くことができません。

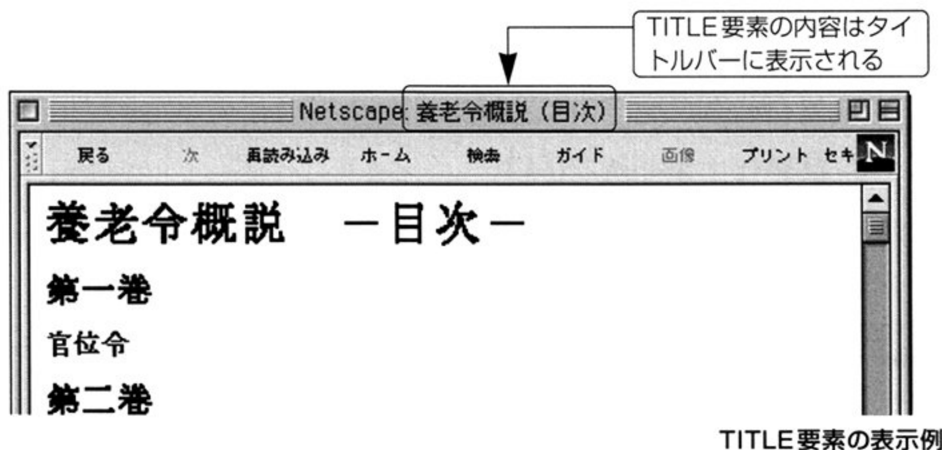
内容はHTML文書全体の内容を端的に表せるものでなければなりません。Webブラウザではウインドウのタイトルバーに表示されますし、「ブックマーク」や「お気に入り」の登録をした際に、その「タイトル」としても使用されます。あるいは、Lynxなどテキストブラウザでは画面上に表示されますし、音声出力の場合にも本文とは区別して読み上げられます。たとえば、ホームページ・リーダーでは「○○（TITLE要素の内容）を読み上げます」と本文の前置きに使用されます。さらに、検索サイトで使用されるサーチロボットも、TITLE要素を「タイトル」として認識し、自己のデータベースに登録します。

なお、TITLE要素を日本語で記述する場合は、若干注意が必要です。というのは、一般的にHTML文書では文字の符号化方式をMETA要素（P.41参照）で明示します。そのため、日本語のTITLE要素がMETA要素の前にあると、Webブラウザによっては正しく日本語として解釈できなくなる可能性もあります。ですから、TITLE要素を日本語で記述する場合は、META要素の後に置くようにしましょう。

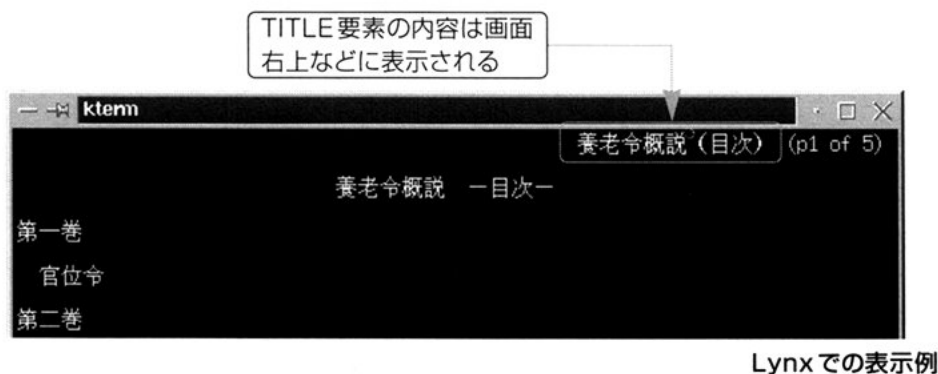
```
<META HTTP-EQUIV="content-type" CONTENT="text/html;
CHARSET=iso-2022-jp">
<TITLE>養老令概説（目次）</TITLE>
```

このHTML文書で使用されている文字の符号化はISO-2022-JPであることを示す（P.309参照）

日本語タイトル



TITLE要素の表示例



Lynxでの表示例

属性一覧

なし

属性解説

なし

ISINDEX

<ISINDEX>（終了タグなし）

キーワード検索（The ISINDEX element）

内容 ▶ 空

親要素 ▶ HEAD

要素の概要

ISINDEX要素は、1行分の入力テキストボックスを表示します。

ユーザがキーワードを入力し、送信用のキー（一般的にはリターンキーなど）を押すと、キーワードが基準URL（BASE要素で設定）のサーバに送信され、サーバ上の他のアプリケーション（CGIスクリプトなど）に引き渡されます。

また、ISINDEX要素は開始タグ<ISINDEX>のみで表現される、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

なお、入力文字数に制限はありませんが、入力可能な文字セットは「ISO 8859-1（欧米の主要14言語をカバーする文字セット）」に限定されます。つまり日本語は使用できません。これは、ISINDEX要素の内容は「記述されるHTML文書の基準URLがHTTP URL（URLがHTTPスキームであること、つまり「http://～」で記述される）であるときのみ使用できる」と定められているためです（URLに使用できる文字セットはISO 8859-1のみ）。そのせいもあって、現在では一般的にISINDEX要素は使用されず、INPUT要素（P.179参照）が使用されています。

属性一覧

●PROMPT

値	働き	H3	N4	I4
text	入力ボックスのプロンプト	○	○	○

属性は省略可

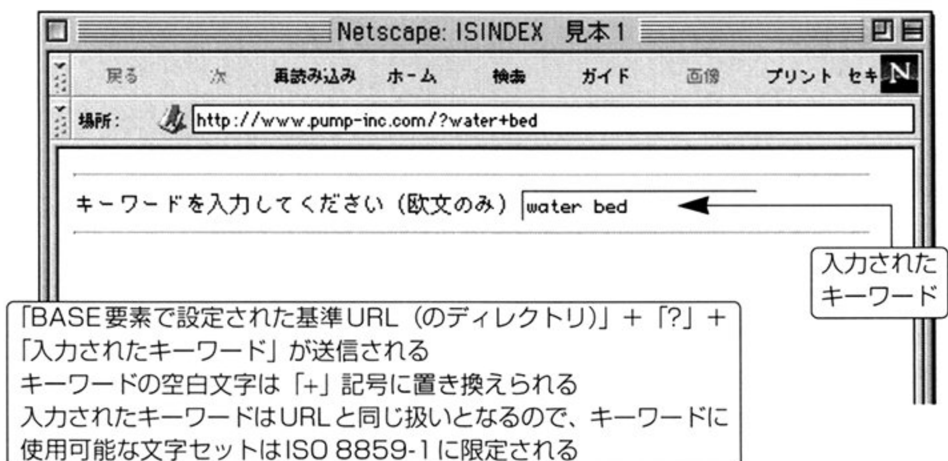
変数　*text*＝一般的なテキスト

属性解説

●PROMPT属性

入力テキストボックスの前に表示させるプロンプトを設定します。

```
<BASE HREF="http://www.pump-inc.com/index.html">
<ISINDEX PROMPT="キーワードを入力してください（欧文のみ）">
```



ISINDEX要素の使用例

BASE

<BASE>（終了タグなし）

基準URL（The BASE element）

内容 ▶ 空

親要素 ▶ HEAD

要素の概要

BASE要素は、そのHTML文書の絶対的な基準となるURLを設定します。以後、同じ文書内の相対URLは、このURLを基準に認識されます。その性格上、できるだけHEAD要素内の一番最初に記述します。またBASE要素を設定しない場合は、Webブラウザは自動的に現在表示している文書のURLを基準URLとして解釈します。

必須の要素ではありませんが、Webサイトの規模が大きい場合などは、ファイルを移動してもリンク関係が維持できるので、メンテナンス性をよくするために使用します。ただし、BASE要素を記述すると、当然リンク先URLとしてHTTPスキーム名を含んだ絶対URLが合成されるため、ローカルのハードディスクでのリンク確認作業ができなくなります（「file:///～」ではなく必ず「http://～」となる）。したがって、一般の個人ユーザは使用しない方がいいかもしれません。

なお、BASE要素は開始タグ<BASE>のみで表現される、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

属性一覧

●HREF

値	働き	H3	N4	I4
url	基準となるURL	○	○	○

属性は必須

●TARGET

値	働き	H3	N4	I4
frame-target	ターゲットにする基準フレーム名	－	！	！

属性は必須



url=URL

frame-target=一般的なテキスト（先頭文字はアルファベットのみ）

属性解説

●HREF属性

BASE要素必須の属性で、基準となるURLを設定します。BASE要素が設定されると、それ以降に記述される相対URLは、全て「基準URL」+「相対URL」の絶対URLとして解釈されます。

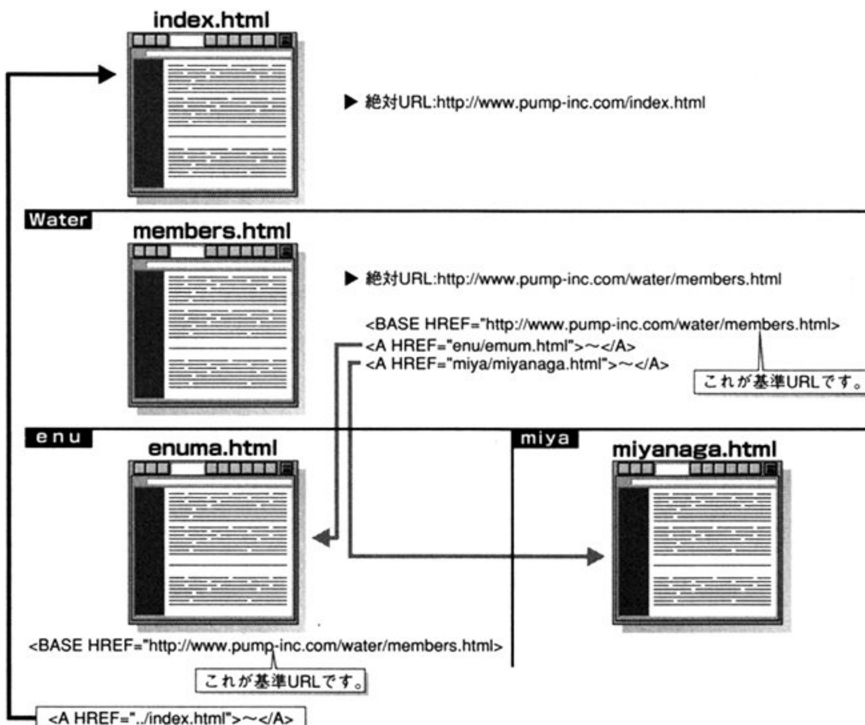
```
<HEAD>
   <BASE HREF="http://www.pump-inc.com/water/
members.html">
</HEAD>
<BODY>
<UL>
 <LI><A HREF="enu/enuma.html">江沼　学（vocal）
</A></LI>
 <LI><A HREF="miya/miyanaga.html">宮永治郎（guitar,
mandolin, chorus）</A></LI>
 <LI><A HREF="kama/kamata.html">鎌田雅人（piano, key
board, chorus）</A></LI>
 <LI><A HREF="oka/okamoto.html">岡本慎吾（bass）
</A></LI>
 <LI><A HREF="yoshi/yoshida.html">吉田太郎（drums）
</A></LI>
 </UL>
（後略）
```

※BASE要素以外のHEAD要素やA要素のTITLE属性などは省略



BASE要素の使用例

注意 HTML文書内でのURLの記述方法には2種類あります。
1つは、スキーム名（転送に使用されるプロトコル名など）部分も含めた完全な形で記述される「絶対URL」です。これはリソースの転送方法や世界中のどこに保管されているかまで、明確に表すために使用されます。もう1つは、スキーム名を記述しない「相対URL」です。こちらは現在表示（出力）されているHTML文書と、同じディスク上に保管されているリソースのありかを示す場合等に使用されます。
なお、基準URLより上位のURL（あるいはそれを経由するURL）を相対URLで記述する場合は、「1つ上のディレクトリ」を表す「..」記号を使用します。



基準URLと相対URL

●TARGET属性

フレーム機能を使用する場合の、ウインドウの表示方法などを定義します(P.263参照)。

NN 3 IE 3 NN 4 IE 4

SCRIPT

<SCRIPT>~</SCRIPT>

スクリプト（The SCRIPT element）

内容 ▶ 一般的なテキスト（本文参照）

親要素 ▶ HEAD　BODY

要素の概要

SCRIPT要素は、使用するクライアントサイド・スクリプト言語（Webブラウザが解釈して実行するスクリプト）を宣言し、スクリプトを記述するために使用します。HEAD要素に属しますが、必要であればBODY要素の中に置くことができ、個数にも制限はありません。

開始タグ<SCRIPT>と終了タグ</SCRIPT>によって、スクリプトをマークアップします。SCRIPT要素の内容は他の要素の内容と異なり、空白文字の連続やタブ、「<」や「&」などの特殊な文字（P.310参照）はそのまま認識されます（文字実体参照で記述すると、それはただの記号として認識される）。そして、最初に「<」「/」の連続（終了タグの開始記号）が現れた時点で、内容の終了として認識されます。

また、SCRIPT要素に対応したWebブラウザは、SCRIPT要素の内容を画面に表示せず、スクリプトとして認識・実行します。しかし、SCRIPT要素がBODY要素に置かれた場合、未対応のWebブラウザではそのまま内容を表示してしまいます。それを防ぐため、一般的にSCRIPT要素の内容は「コメント」として記述されます。

なお、HTML 3.2では要素名が予約されただけで、正式に採用されたものではありません。実際にSCRIPT要素を使用する場合は、HTML 3.2ではなくHTML 4.0の規格に則ってHTML文書を作成する必要があります。スクリプトの概要はP.246で触れますが、詳細は別途HTML 4.0やスクリプトの専門解説書をご覧ください。

属性一覧

なし

属性解説

なし

STYLE

<STYLE>~</STYLE>

スタイル（The STYLE element）

内容 ▶ 一般的なテキスト（本文参照）

親要素 ▶ HEAD

要素の概要

STYLE 要素は、文書内で使用されるスタイルシートをまとめて設定するために使用します。HEAD 要素内に置く必要がありますが、個数に制限はありません。

開始タグ<STYLE>と終了タグ</STYLE>によって、スタイルシートをマークアップします。また、STYLE 要素の内容は他の要素の内容と異なり、空白文字の連続やタブ、「<」や「&」などの特殊な文字（P.310 参照）がそのまま認識されます（文字実体参照で記述すると、それはただの記号として認識される）。そして、最初に「<」「/」の連続（終了タグの開始記号）が現れた時点で、内容の終了として認識されます。

また、STYLE 要素は必ずHEAD 要素内に置かれるため、Web ブラウザの対応・未対応に関わらず内容が表示されることはないはずですが、慎重を期して「コメント」として記述する人もいるようです。

なお、HTML 3.2 では要素名が予約されただけで、正式に採用されたものではありません。実際にSTYLE 要素を使用する場合は、HTML 3.2 ではなくHTML 4.0 の規格に則ってHTML 文書を作成する必要があります。詳細は別途HTML 4.0 やスタイルシートの専門解説書をご覧ください。

属性一覧

なし

属性解説

なし

META

<META>（終了タグなし）

文書情報（The META element）

内容 ▶ 空

親要素 ▶ HEAD

要素の概要

META要素は、文書に関するさまざまな基礎情報、たとえば文書の作者や有効期限、キーワードなどを記述するために使用します。情報は通常「プロパティ名の属性＋プロパティ値の属性」のセットで表現されます。

また、META要素は私たち日本語使用者（というより、非西欧言語使用者）にとっては、文書内で使用される文字の符号化方式の明示にも使用できる大変重要な要素です。ただし、個々の値に対するサーバやWebブラウザなどクライアント側の動作についての細かい規定はなく、具体的にどのように動作するか、あるいは無視するかは各ソフトウェアに任されています。

なお、META要素は開始タグ<META>のみで表現される、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

属性一覧

●HTTP-EQUIV

値	働き	H3	N4	I4
name	情報名の設定	○	○	○

属性は省略可

●NAME

値	働き	H3	N4	I4
name	情報名の設定	○	○	○

属性は省略可

●CONTENT

値	働き	H3	N4	I4
data	具体的な値の設定	○	○	○

属性は必須



name=アルファベット、数字、ハイフン、ピリオド（先頭はアルファベットのみ
NAME属性は大文字小文字の区別あり）

data=一般的なテキスト（大文字小文字の区別あり　パラメータ値は除く）

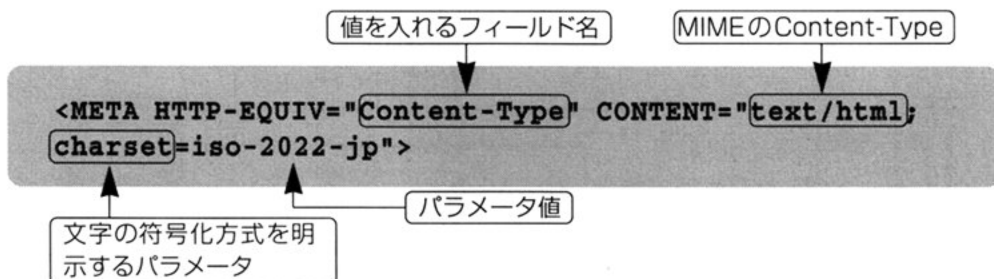
属性解説

●HTTP-EQUIV属性

HTMLサーバがHTTPヘッダで使用するフィールド名を設定します。値の先頭(名前開始文字)は「アルファベット」のみ、続く文字列(名前文字)は「アルファベット、数字、ハイフン、ピリオド」に限定されます。(P.8参照)

HTML文書内にMETA要素とHTTP-EQUIV属性があると、サーバはWebブラウザなどのアプリケーションから要求を受けてHTML文書を送信する際、HTTPヘッダの指定されたフィールドにCONTENT属性で設定された値を入れて送信します。

たとえば、HTML文書の符号化方式がISO-2022-JP(通称JISコードP.309参照)である場合、以下のようにMETA要素を記述します。

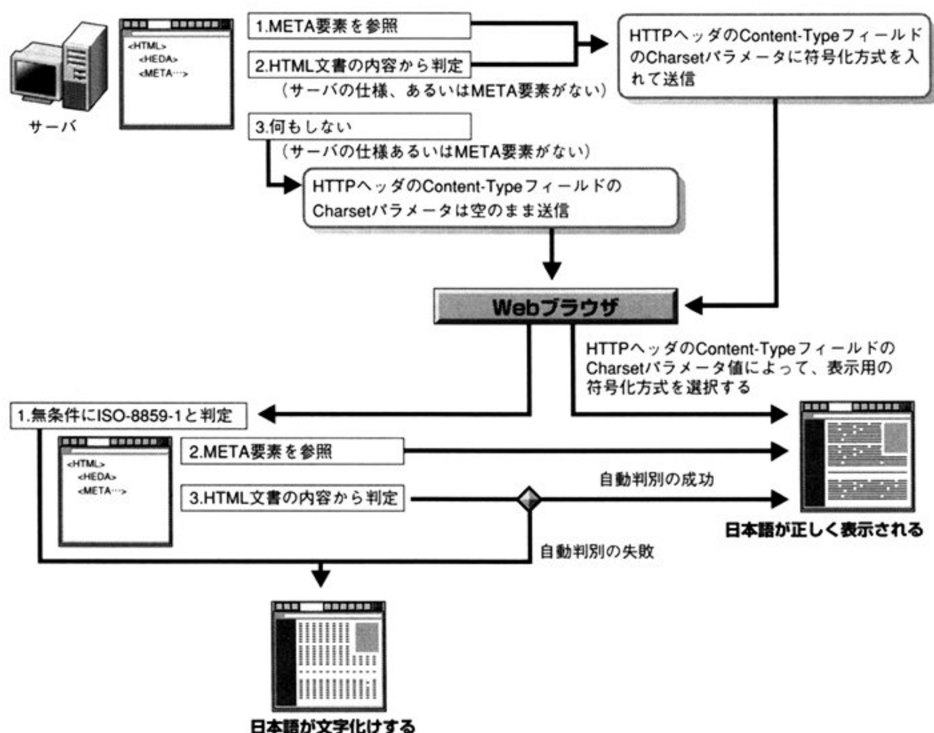


Webブラウザからの HTML文書の送信要求を受けたサーバは、まずHTML文書内にMETA要素とHTTP-EQUIV属性があるかどうかチェックします。上記の例では、サーバはHTTPヘッダのContent-Typeフィールドに「text/html; charset=iso-2022-jp」という値を入れて、HTML文書と一緒にWebブラウザへ送信します。この記述方法は、MIME(P.59コラム参照)のContent-Typeフィールドと同じです。

受信したWebブラウザは、HTTPヘッダの「Content-Type」フィールドの値を見ただけで、そのHTML文書の符号化方式がISO-2022-JPであることがわかりますので、画面表示や印刷の際の符号化方式を間違いなく選択できます。

ところが、サーバによっては送信前にMETA要素をチェックせず、符号化方式をパラメータ値に入れてくれないサーバもあります。もし、符号化方式が含まれていない場合は、HTTPの規定では、Webブラウザは「符号化方式はISO-8859-1」と判断するよう求められています。もちろん、これでは日本語のHTML文書を表示することはできません。

そこで、Webブラウザの多くは、HTTPヘッダのContent-Typeフィールドのパラメータ値が抜けていても、受信したHTML文書のMETA要素をチェックして、符号化方式を判定する機能を備えています。そのため、HTML文書内のMETA要素に符号化方式が記述されていれば、表示の際の文字化けは防げるのです（古いWebブラウザでは対応していないものもありますが…）。



符号化方式判定の流れ

●HTML文書の文字コード判定

場所:	http://www.pump-inc.com/META1.html
ファイルMIMEタイプ:	text/html
ソース:	現在、ディスクキャッシュに入っています
ローカル キャッシュ ファイル:	cache1129280.html
最終更新日時:	不明
最終更新日時:	不明
内容サイズ:	654
有効期限:	日付なし
文字セット:	iso-2022-jp
セキュリティ:	このドキュメントでは暗号化によるセキュリティ保護は行われていません。

符号化方式がISO-2022-JPと判別されている

Netscape Navigatorの「ドキュメント情報」の表示例

では、HTML文書のMETA要素に符号化方式の記述がなかったらどうなるでしょうか？　まず、サーバはHTTPヘッダのContent-Typeフィールドのパラメータ値に符号化方式を含めることができません。サーバによっては、META要素の有無に関係なく、専用の辞書を使用して文書内容をチェックし、符号化方式を判定するものもあります(中には文書のほんのさわりだけをチェックして判定を行う横着なサーバもあるようです)。これは一見親切なようですが、残念ながらこの判定は常に正しいとは限りません。

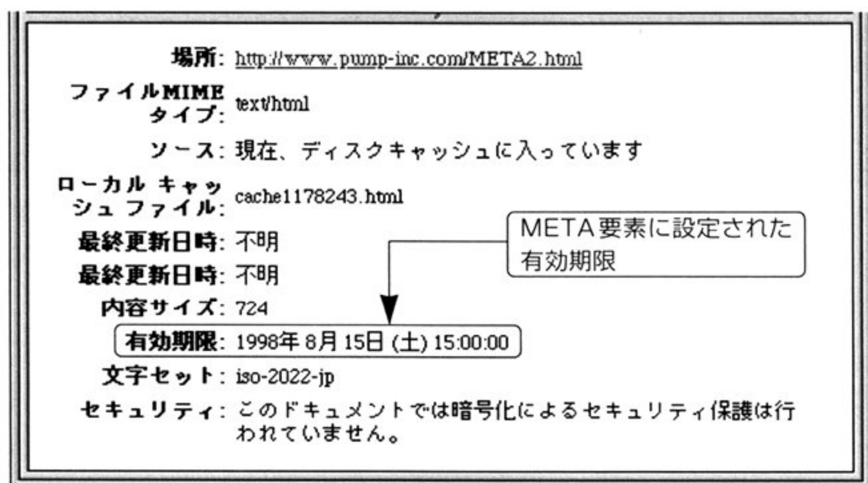
また、HTML文書を受信したWebブラウザは、仕方ないので独自の方法で符号化方式の判定を試みます。これが現在多くのWebブラウザに搭載されている「文字コードの自動判別機能」ですが、ご存じの通りこれはよく失敗します。特に日本語の場合、EUC（主にUNIXで使用される符号化方式）とシフトJIS（Mac OSやWindows95などパソコンで使用される符号化方式）の判定は困難なようです。

以上、長くなりましたが、非西欧言語使用者にとってMETA要素がいかに重要であるかおわかりいただけたかと思います。

その他、一般的に使用されるHTTP-EQUIV属性値（すなわち、HTTPヘッダのフィールド）には、「Expires」があります。これは指定された日時で有効期限が切れることを知らせます。

```
<META HTTP-EQUIV="Expires" CONTENT="Sat, 15 Aug 1998
15:00:00 JST">
```

上記の例では、これを受けたWebブラウザは「この文書は1998年8月15日15時（日本標準時）で有効期限が切れるので、それ以降はキャッシュからではなくオリジナルのデータをサーバから受信する」よう判断します。なお、HTTPヘッダに関する詳細は、HTTPの解説書をご覧ください。



Netscape Navigatorの「ドキュメント情報」の表示例

注意 もともとHTTP-EQUIV属性はサーバのための属性でしたが、Webブラウザが符号化方式の判定以外にも積極的に利用することもあります。
たとえば、「Refresh」という値を用いると、指定した時間後に別のURLへ自動的に移動するような指示をWebブラウザに対して行うことができます。

```
<META HTTP-EQUIV="Refresh" CONTENT="5;
URL=http://www.pump-inc.com/index.html">
```

上記の例では、5秒後に「http://www.pump-inc.com/index.html」に自動的に移動します。使い方によっては紙芝居的なWebページを作成することも可能です。

●NAME属性

NAME属性も情報名の設定に使用されます。

サーチエンジンなどでは、META要素の中に記述されたキーワードを頼りに検索を行ったり、索引を作成したりするものがあります。具体的なNAME属性の値は規定がありませんが、使用される文字については、先頭(名前開始文字)は「アルファベット」のみ、続く文字列(名前文字)は「アルファベット、数字、ハイフン、ピリオド」に限定されます。一般的には「Keyword」などが使用されています。

```
<META NAME="Keyword" CONTENT="animal, cat, dog">
```

HTML文書のキーワードとなる単語

また、最近では検索サイトのサーチロボットのチェックに対して制限をかける場合もMETA要素が使用されています。その場合のNAME属性の値として、「ROBOTS」があります。

```
<META NAME="ROBOTS" CONTENT="NOINDEX">
```

上記の例では、CONTENT属性で具体的に「索引付け拒否」を宣言しています。他にもリンク内容のチェックを拒否する「NOFOLLOW」などがあります。ただし、全てのサーチロボットがMETA要素に対応しているわけではありません。

●CONTENT属性

CONTENT属性は、HTTP-EQUIV属性やNAME属性の具体的な値を設定します。

使用例はHTTP-EQUIV属性とNAME属性の項をご覧ください。

NN 3 IE 3 NN 4 IE 4

LINK

<LINK>（終了タグなし）

文書情報（The LINK element）

内容 ▶ 空

親要素 ▶ HEAD

要素の概要

LINK要素は、文書の作者への連絡先を表したり、複数の文書（リソース）間の関係（読み込みや印刷の順序など）、あるいは外部スタイルシートを指定するために使用します。必ずHEAD要素内に置く必要がありますが、その個数に実用上の制限はありません。

直接画面表示などに関わる部分が少なく、またHTML 3.2ではスクリプトやスタイルシートに対応していないこともあって、積極的に使用されることの少ない要素ですが、対応しているWebブラウザやアプリケーションのユーザにとっては重要な要素です。Netscape NavigatorやInternet Explorerではその効果がわかりにくいと思いますが、できれば記述するよう試みてください。

なお、LINK要素は開始タグ<LINK>のみで表現される、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

属性一覧

●HREF

値	働き	H3	N4	I4
url	リンク先リソースのURL	○	○	○

属性は省略可

●REL

値	働き	H3	N4	I4
link type	正順の関係の種類	○	○	○

属性は省略可

●REV

値	働き	H3	N4	I4
link type	逆順の関係の種類	○	○	○

属性は省略可

●TITLE

値	働き	H3	N4	I4
text	リンク先リソースのタイトル	○	−	−

属性は省略可

●TYPE

値	働き	H3	N4	I4
type	外部スタイルシートの種類	−	!	!

属性は省略可


url=URL　*link type*=文書間の関係の種類　*type*=スタイルシートの形式
text=一般的なテキスト

属性解説

●HREF属性

リンク関係が設定されたリソース（HTML文書とは限りません）のURLを指定します。

具体的なリンク関係の種類については、REL・REV属性で設定します。

●REL属性

当該HTML文書からHREF属性で指定されたリンク先リソース（HTML文書とは限りません）を見た場合の関係の種類を設定します。LINK要素はHEAD要素の1つなので、REL属性の内容もHTML文書の「本文」としては表示されません。しかし、Webブラウザによってはハイパーリンク・ナビゲーションシステムとして、別途表示されます。あるいは、リンク先のリソースをWebブラウザにバックグラウンドでリンク先文書にアクセスさせ（いわゆるデータの先読み）、表示時間の短縮を図るような使い方も想定できます。

なお、HTML 3.2であらかじめ規定されている値（関係の種類）には、以下のようなものがあります。

値	内容
top：	トップページ（表紙）など
contents：	目次ページなど
index：	索引ページなど
glossary：	用語解説ページなど
copyright：	著作権に関する情報ページなど
next：	当該文書の次に移動すべきページなど
previous：	当該文書の前のページなど

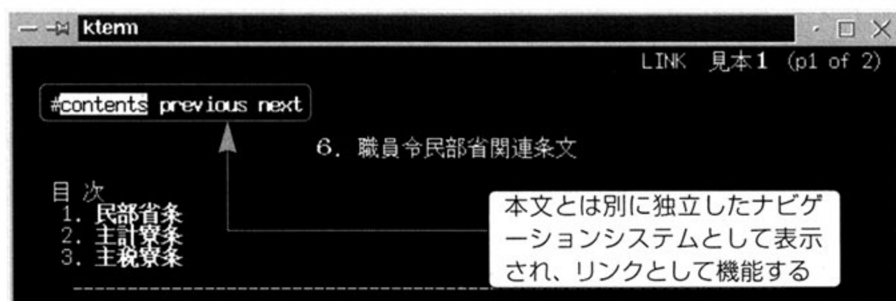
値	内容
help：	ヘルプページなど
search：	検索ページなど

```
<LINK REL="contents" HREF="http://www.heian.go.jp
/mokuji.html">
<LINK REL="previous" HREF="http://www.heian.go.jp
/shikiinn/05wosamuru.html">
<LINK REL="next" HREF="http://www.heian.go.jp
/shikiinn/07tsuhamono.html">
```

当該HTML文書の目次ページ
前のページ
次のページ



LINK要素の使用例



Lynxでの表示例

●REV属性

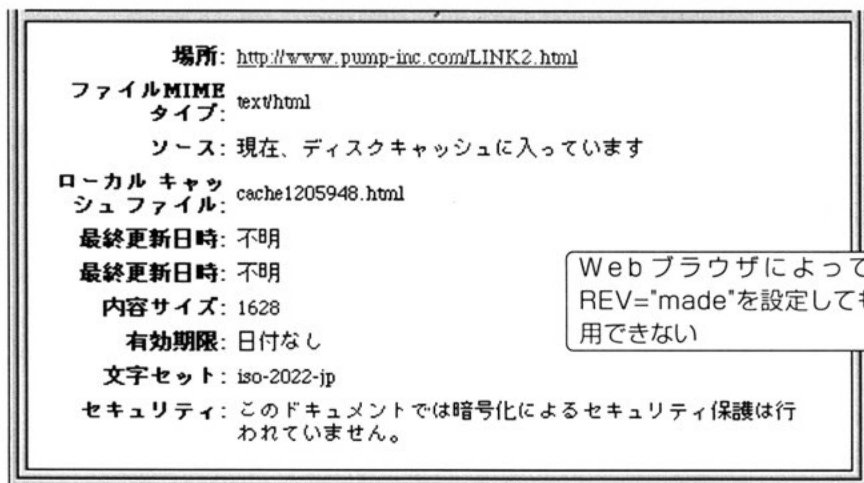
当該HTML文書を、HREF属性で指定されたリンク先リソース（HTML文書とは限りません）から見た場合の関係の種類を設定します（REL属性と逆方向の関係を表す）。

HTML 3.2で規定されている値はREL属性と同じですが、REV属性はそれ以外に「made」という値が使用することができます。これは当該文書の作者の連絡先な

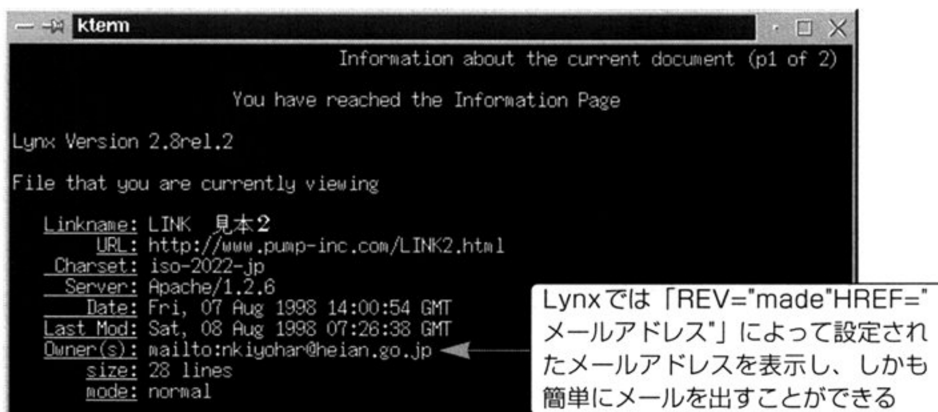
どを明示するための値で、HREF属性には作者のメールアドレスなどを記述します。Webブラウザによってはhtml文書の持つ情報としてこの内容を表示し、簡単に作者へメールが出せる機能を持つものもあります。

```
<LINK REV="made" HREF="mailto:nkiyohar@heian.go.jp">
```

作者のメールアドレス



Netscape Navigatorの「ドキュメント情報」の表示例



Lynxの「文書情報」表示例

●TITLE属性

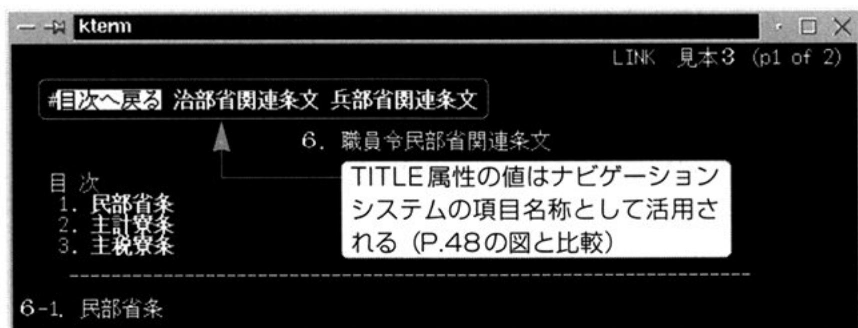
HREF属性で設定したリンク先リソースのタイトル（情報）を設定しますが、実際の表現方法に関しては、Webブラウザに任されています。

たとえば、LynxではREL属性で表示されるナビゲーションシステムの項目名称に使用されます。

```
<LINK TITLE="目次へ戻る" REL="contents" HREF=
"http://www.heian.go.jp/mokuji.html">
<LINK TITLE="治部省関連条文" REL="previous" HREF=
"http://www.heian.go.jp/shikiinn/05wosamuru.html">
<LINK TITLE="兵部省関連条文" REL="next" HREF=
"http://www.heian.go.jp/shikiinn/07tsuhamono.html">
<LINK REV="made" HREF="mailto:nkiyohar@heian.go.jp">
```



TITLE属性の使用例



Lynxでの表示例

●TYPE属性

外部スタイルシートの種類を指定します。詳細は別途HTML4.0かスタイルシートの専門解説書をご覧ください。

Hn (H1,H2,H3,H4,H5,H6) <Hn>~</Hn>

見出し (Headings)　　ブロックレベル要素

内容 TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET IMG FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

親要素 BODY
DIV CENTER BLOCKQUOTE FORM TH TD

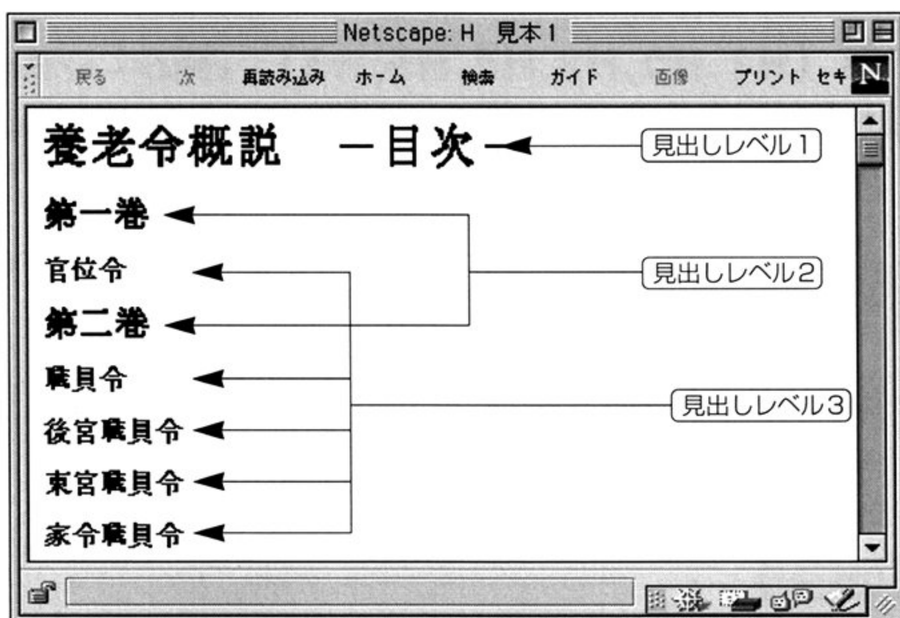
要素の概要

見出し要素は、HTML文書本文の見出しに使用します。開始タグ<Hn>と終了タグ</Hn>によって見出しをマークアップします。nは1～6までの数値で、見出しのレベル（重要度）を表し、H1は最も重要度が高い見出しで、H6は最も重要度が低い見出しになります。

また、見出しはブロックレベル要素の一種ですが、その内容にはテキストレベル要素と一般的なテキストのみ置くことができます。見出しを入れ子にしたり、他のブロックレベル要素を置くことはできません。

Webブラウザでは一般的に見出しは太字で表示されます。また、数値が大きくなるほど、即ち重要度が下がるにつれて文字が小さく表示されます。しかし、これはWebブラウザがそう表示しているだけなので、それを頼りに見出し要素を文字飾りのために使用してはいけません。たとえば、多くの検索サイトで使用されているサーチロボットは、自動的にチェックしたHTML文書の概要を作成する際にこの見出し要素を利用します。そのため、いい加減な見出しをつけていると、検索サイトで自分のWebページを正しく紹介してもらえなくなります。

```
<H1>養老令概説　一目次一</H1>          ← 見出しレベル1
    <H2>第一巻</H2>                    ← 見出しレベル2
        <H3>官位令</H3>                ← 見出しレベル3
    <H2>第二巻</H2>                    ← 見出しレベル2
        <H3>職員令</H3>                ← 見出しレベル3
        <H3>後宮職員令</H3>            ← 見出しレベル3
        <H3>東宮職員令</H3>            ← 見出しレベル3
        <H3>家令職員令</H3>            ← 見出しレベル3
…（以下略）
```



H1からH3要素の使用例

なお、テキストブラウザでは文字の飾りやサイズが変更できないので、表示位置の調整（たとえばH1要素が中央揃え、H3要素がインデント）で表現することもあります。

注意 一般的にH1要素は文書の見出し≒タイトルとして使用されていますが、HTML文書全体のタイトル（門柱の表札）を示すTITLE要素（P.31参照）と異なり、H1要素はHTML文書本文（屋内）の一要素です。検索エンジンは、H1要素を「タイトル」ではなくあくまでも「見出し」として認識しますし、閲覧者がWebブラウザでブックマークした場合にもH1要素の内容は反映されません。TITLE要素とH1要素の本質的な役割の違いには、注意が必要です。

属性一覧

●ALIGN（初期値：left）

値	働き	H3	N4	I4
left	左揃え	○	○	○
center	中央揃え	○	○	○
right	右揃え	○	○	○

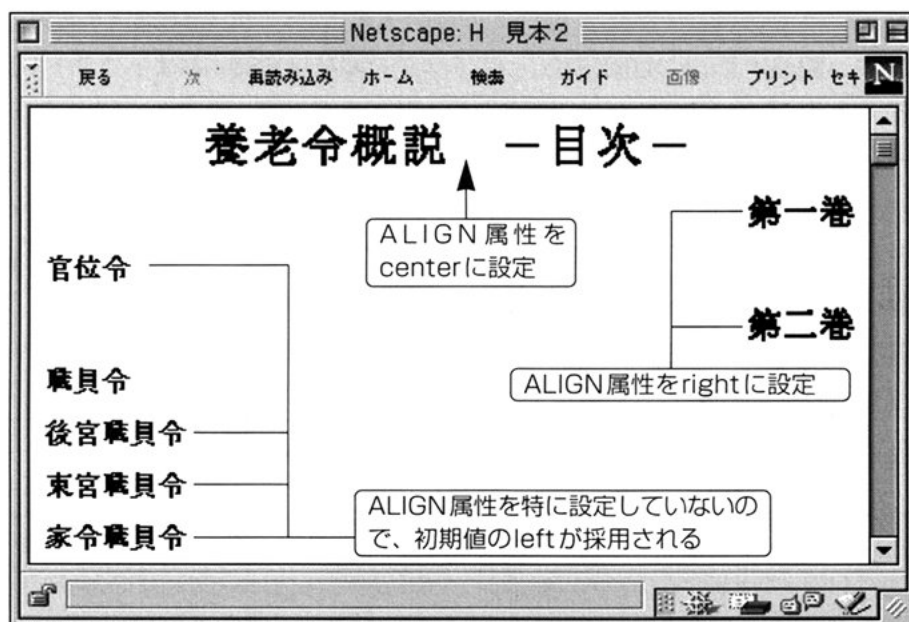
属性は省略可

属性解説

●ALIGN属性

ALIGN属性は、見出しの行揃え位置を指定します。属性値を省略すると通常は「ALIGN="left"」が適用されます。

```
<H1 ALIGN="center">養老令概説　一目次一</H1>
     <H2 ALIGN="right">第一巻</H2>
          <H3>官位令</H3>
     <H2 ALIGN="right">第二巻</H2>
          <H3>職員令</H3>
          <H3>後宮職員令</H3>
          <H3>東宮職員令</H3>
          <H3>家令職員令</H3>
```



ALIGN属性を指定した例

NN 3 IE 3 NN 4 IE 4

ADDRESS <ADDRESS>～</ADDRESS>

アドレス（The ADDRESS element） ブロックレベル要素

内容 ▶ P
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 ▶ BODY
DIV CENTER BLOCKQUOTE FORM TH TD

要素の概要

ADDRESS要素は、HTML文書作者の連絡先や内容の問い合わせ先の記述に使用します。開始タグ<ADDRESS>と終了タグ</ADDRESS>によってアドレスをマークアップします。通常はHTML文書の末尾や先頭に置かれます。

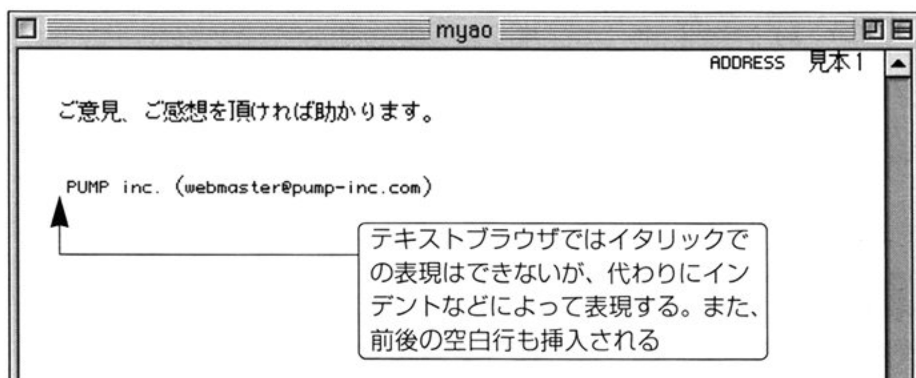
また、ブロックレベル要素の一種ですが、ADDRESS要素を入れ子にしたり、他のブロックレベル要素を入れることはできず、テキストレベル要素と一般的なテキストのみ内容として置くことができます（例外的にブロックレベル要素のうちP要素のみ置くことができますが、あまり使用する必要性はないでしょう）。

通常、ADDRESS要素はイタリック表示されます。また、HTML 3.2ではWebブラウザでの表示の際には、前後に空白行が挿入されることを推奨していますが、実際には多くのWebブラウザでは空白行は挿入されません。

```
ご意見、ご感想を頂ければ助かります。
<ADDRESS>PUMP inc. (webmaster@pump-inc.com) </ADDRESS>
```



ADDRESS要素の表示例



Lynxでの表示例

属性一覧

なし

属性解説

なし

P

<P>~</P>

段落（Paragraphs） **ブロックレベル要素**

内容 TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

親要素 BODY
ADDRESS
DIV CENTER BLOCKQUOTE FORM TH TD DD LI

要素の概要

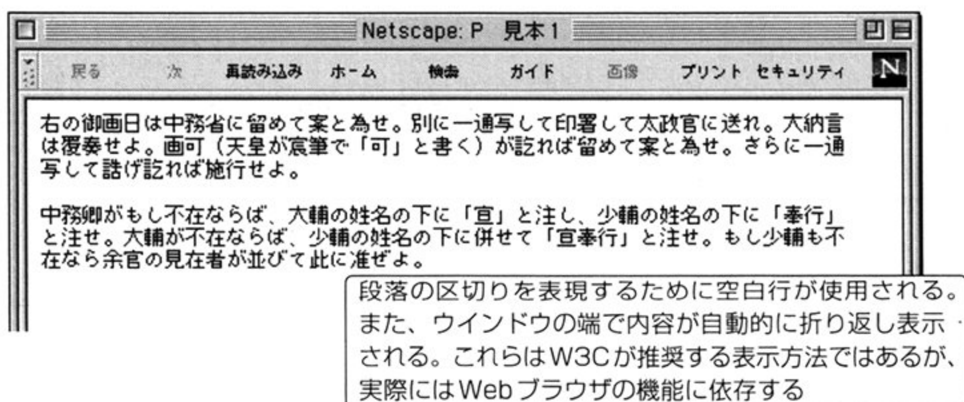
P要素は、HTML文書本文の段落を示す要素です。

開始タグ<P>と終了タグ</P>によって、１つの段落をマークアップします。終了タグ</P>は省略することができますが、記述しておいた方が後々段落の区切りが明確にわかるので、修正なども楽になると思います。

P要素の内容には、テキストレベル要素と一般的なテキストを置くことができます。もちろん、画像やJava Appletなどマルチメディアデータも含めることができます。というより、含める方が文書の構造として自然です（P.10参照）。一方、P要素を入れ子にしたり、他のブロックレベル要素を置くことはできませんので、注意してください。

注意 開始タグ<P>の後ろに何らかのブロックレベル要素が現れると、その直前に終了タグ</P>の存在が自動的に推定されます。そのため、終了タグ</P>の省略が可能となります。

<P> 右の御画日は中務省に留めて案と為せ。別に一通写して印署して太政官に送れ。大納言は覆奏せよ。画可（天皇が宸筆で「可」と書く）が訖れば留めて案と為せ。さらに一通写して誥げ訖れば施行せよ。**</P>**
<P> 中務卿がもし不在ならば、大輔の姓名の下に「宣」と注し、少輔の姓名の下に「奉行」と注せ。大輔が不在ならば、少輔の姓名の下に併せて「宣奉行」と注せ。もし少輔も不在なら余官の見在者が並びて此に准ぜよ。**</P>**



P要素の表示例

P要素が表示の際に作成する空白行は、「段落の開始行頭をインデントしない場合は、段落を空白行で区切って明示する」という英文の段落表現の1つです。しかし、一般の英文では段落の開始は行頭のインデントで表現しますので、英文を見慣れた人にも違和感があるでしょう。

このような段落表現は、初期のWebブラウザにハイフネーションやワードラップの機能がなく、インデントによって満足に段落を表現することができなかったことに起因しています。古いHTML文書作成ツールの中には、この特性を利用して空白行を作るために<P>タグを挿入するものがありますが、現在では多くのWebブラウザは内容のないP要素を無視しますし、文書の構造上も矛盾していますので、このようなP要素の使用方法は避けてください。

属性一覧

●ALIGN（初期値：left）

値	働き	H3	N4	I4
left	左揃え	○	○	○
center	中央揃え	○	○	○
right	右揃え	○	○	○

属性は省略可

属性解説

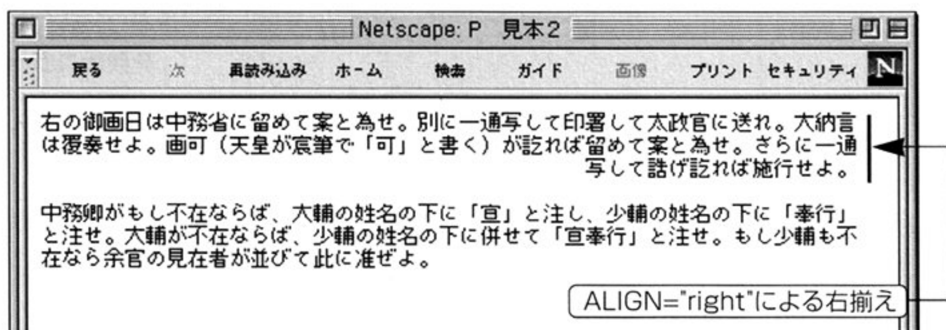
●ALIGN属性

ALIGN属性は段落の行揃え位置を指定します。

属性を省略すると初期値leftが与えられ、自動的に左揃えになります。

ALIGN属性の指定例

```
<P ALIGN="right">右の御画日は中務省に留めて案と為せ。別に一通写
して印署して太政官に送れ。大納言は覆奏せよ。画可（天皇が宸筆で「可」と
書く）が訖れば留めて案と為せ。さらに一通写して誥げ訖れば施行せよ。</P>
<P>中務卿がもし不在ならば、大輔の姓名の下に「宣」と注し、少輔の姓名の
下に「奉行」と注せ。大輔が不在ならば、少輔の姓名の下に併せて「宣奉行」
と注せ。もし少輔も不在なら余官の見在者が並びて此に准ぜよ。</P>
```

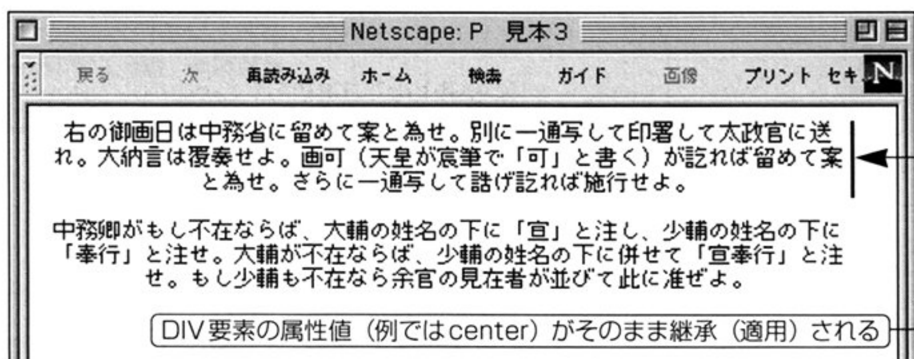


ALIGN属性の使用例

また、DIV要素やCENTER要素の内容となっている場合は、属性を省略するとそちら（親要素）の属性値が適用されます。

DIV要素で中央揃えを指定

```
<DIV ALIGN="center">
<P>右の御画日は中務省に留めて案と為せ。別に一通写して印署して太政官に
送れ。大納言は覆奏せよ。画可（天皇が宸筆で「可」と書く）が訖れば留めて
案と為せ。さらに一通写して誥げ訖れば施行せよ。</P>
<P>中務卿がもし不在ならば、大輔の姓名の下に「宣」と注し、少輔の姓名の
下に「奉行」と注せ。大輔が不在ならば、少輔の姓名の下に併せて「宣奉行」
と注せ。もし少輔も不在なら余官の見在者が並びて此に准ぜよ。</P>
</DIV>
```



DIV要素の使用例

Column　MIMEについて

インターネットは米国生まれのため、メールシステムも基本的にはASCII文字セットしかサポートしていません。それでは英語以外の言語使用者は困りますし、テキスト以外の画像や動画となるとなおさらです。そこで、考案されたのがMIME（Multipurpose Internet Mail Extensions）という仕組みで、メールのヘッダにさまざまなフィールドを付加してメールの情報を明示し、アプリケーションが取り扱いに困らないようにします。

このMIMEの仕組みはメールシステムの他にも、いろいろな場面で使用されます（HTML文書を転送ためのプロトコルであるHTTPなど）。なお、以下に簡単にフィールドとその値の記述例を示しますが、詳細については、別途ネットワーク関連の解説書をご覧ください。

・MIME-Version: 1.0
　MIMEのバージョン。
・Content-Type: text/plain; charset=iso-2022-jp
　受信側のアプリケーションがデータを元通りに復元するための情報が「データタイプ／サブタイプ」の形式で記述されます。また、文字の符号化方式や添付ファイルの名前、その他の情報がパラメータとして付加されます。
・Content-Transfer-Encoding: 7bit
　データ転送時の変換方式。MIMEに対応したメールソフトは、通常送信時に元の文字を全てASCII文字に変換しますので「7bit」と記述されます。また、添付ファイルでは変換方式が記述されます。
・Contend-ID: <id4542850590@heian.go.jp>
　メールの識別用のIDです。
・Content-Description: inline; filename="waterbed.jpg"
　添付ファイルがある場合などに、データの説明が記述されます。

BR

**
（終了タグなし）**

強制改行（force a line BRake） テキストレベル要素

内容 ▶ 空

親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DIV CENTER BLOCKQUOTE FORM CAPTION
TH TD DT DD LI
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
APPLET FONT

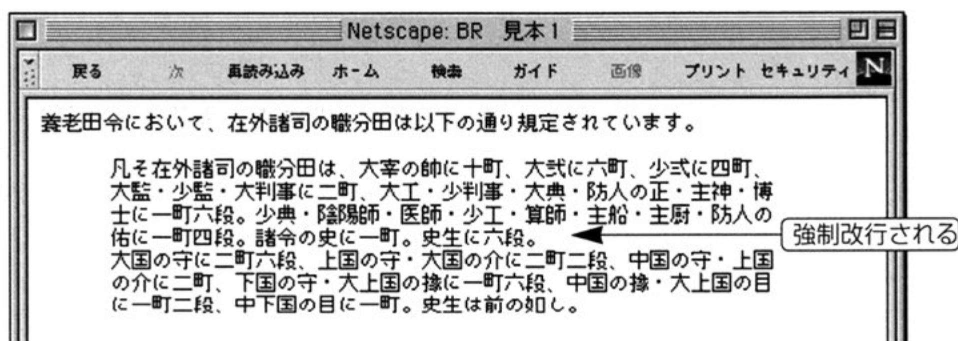
要素の概要

BR要素は、強制改行のために使用される要素です。

HTML文書では、リターンキーを押してテキストを改行しても、Webブラウザなどアプリケーションはそれを無視するよう決められています。したがって、必要があって改行を表現したい場合には、BR要素を用います。たとえば、長すぎる段落を読みやすくするために形式的な段落を作成する場合、あるいは段落内で箇条書きしたい場合などに使用します。

なお、BR要素は開始タグ
のみでマークアップされる、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

```
<P>養老田令において、在外諸司の職分田は以下の通り規定されています。
</P>
<BLOCKQUOTE>凡そ在外諸司の職分田は、大宰の帥に十町、大弐に六町、少
弐に四町、大監・少監・大判事に二町、大工・少判事・大典・防人の正・主
神・博士に一町六段。少典・陰陽師・医師・少工・算師・主船・主厨・防人
の佑に一町四段。諸令の史に一町。史生に六段。<BR> ←開始タグ
大国の守に二町六段、上国の守・大国の介に二町二段、中国の守・上国の介
に二町、下国の守・大上国の掾に一町六段、中国の掾・大上国の目に一町二
段、中下国の目に一町。史生は前の如し。</BLOCKQUOTE>
```



BR要素の使用例

属性一覧

●CLEAR（初期値：none）

値	働き	H3	N4	I4
left	テキストの回り込みを解除（左側）	○	○	○
right	テキストの回り込みを解除（右側）	○	○	○
all	テキストの回り込みを解除（全部）	○	○	○
none	テキストの回り込みを制御しない	○	×	×

属性は省略可

属性解説

●CLEAR属性

画像に対してテキストの回り込みがある場合（IMG要素のALIGN属性）、CLEAR属性を設定することによって、改行と同時に回り込みを解除することができます。

属性値leftは左側の画像に対する回り込みを解除し、rightは右側の画像に対する回り込みを解除します。また、allは両側の画像に対する回り込みを解除します。

なお、属性値noneを設定した場合、Netscape NavigatorとInternet Explorerは解釈できません。しかし、属性を省略することによって、その機能は実現できますので、実用上は問題はありません。

```
<P>
<IMG SRC="wti.jpg" ALIGN="left">コイツは、手がつけられな
い！！<BR>
全曲シングルかと思わせるクオリティ<BR>
他のバンドとはひとまわり違うスケール<BR CLEAR="left">
R&amp;B をベースにした実力派ブリットポップバンド <B>water</B> の
1st アルバム「奇跡の人」(TOCT-10271)
</P>
```

CLEAR属性の指定



CLEAR属性によるテキスト回り込みの回避

UL

<UL>~</UL>

順不同リスト（Unordered List）　ブロックレベル要素

内容 ▶ LI
※1個以上

親要素 ▶ BODY
DIV　CENTER　BLOCKQUOTE　FORM　TH　TD　DD
LI

要素の概要

UL要素は、順不同リストの要素です。主に順番がそれほど重要でないリスト（単なる箇条書きなど）に用います。

開始タグ<UL>と終了タグ</UL>によって、具体的なリスト項目をマークアップします。その中には必ず1個以上のLI要素（P.71参照）が含まれます。LI要素以外の要素を含むことはできません。

なお、HTML 3.2にはUL要素と似た機能を持つDIR・MENU要素もあります。DIR要素は階層型リスト、MENU要素はメニュー項目リストの要素です。HTML 3.2では、Webブラウザはそれらしく表示することを求めていますが、ほとんどのWebブラウザはUL要素全く同一に解釈・表示します（現在使用されることはまずありません）。

```
<P>孟夏条</P>
<UL>                          ← 開始タグ
    <LI>神衣祭</LI>  ←┐
    <LI>大忌祭</LI>  ←┤
    <LI>風神祭</LI>  ←┼─ リスト項目
    <LI>三枝祭</LI>  ←┘
</UL>                         ← 終了タグ
```

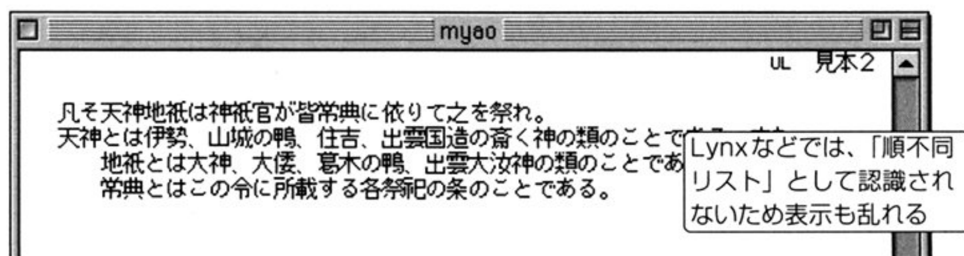


UL要素の使用例

なお、Webブラウザでは前ページのサンプル画像のようにLI要素がインデントして表示されますが、この表示方法はHTMLで定義されているわけではありません。したがって、環境やWebブラウザによって見え方が異なる可能性もあります。

注意 HTMLにはテキストをインデントさせるための要素はありませんが、文書の内容によってはどうしてもインデントが必要なことがあるかもしれません。一部のHTML文書作成ツールでは、<UL>～</UL>タグでテキストをマークアップしてインデントを表現しようとしますが、Netscape NavigatorやInternet Explorerではインデント表示されても、他のWebブラウザでもそうなるとは限りません。たとえば、Lynxでは単に<UL>～</UL>タグでマークアップしただけのテキストは表示が乱れます。また、Netscape NavigatorやInternet Explorerでも将来要素の解釈が厳密になった場合、インデントされなくなる可能性があります。

```
<P>凡そ天神地祇は神祇官が皆常典に依りて之を祭れ。</P>
<UL>天神とは伊勢、山城の鴨、住吉、出雲国造の斎く神の類のことである。
また、地祇とは大神、大倭、葛木の鴨、出雲大汝神の類のことである。さらに、
常典とはこの令に所載する各祭祀の条のことである。</UL>
```



UL要素のみでマークアップした例

属性一覧

●TYPE（初期値：disc）

値	働き	H3	N4	I4
disc	行頭記号を●で表示	○	○	○
square	行頭記号を□で表示	○	○	○
circle	行頭記号を○で表示	○	○	○

属性は省略可

●COMPACT（論理型）

値	働き	H3	N4	I4
compact	リストの短縮表示	○	×	×

属性は省略可

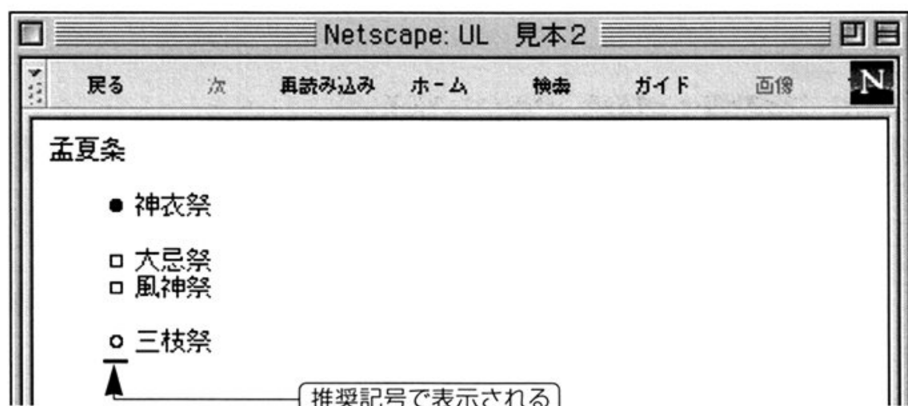
属性解説

●TYPE属性

Webブラウザが表示する行頭記号の種類を指定します。HTMLではなるべくdiscは●、squareは□、circleは○で表示することを推奨していますが、強制ではありません。したがって、環境やWebブラウザによって表示される記号が異なったり、区別できない場合もあります。あまり厳密な再現性は期待しない方がいいでしょう。

```
<P>孟夏条</P>
    <UL TYPE="disc">
        <LI>神衣祭</LI>
    </UL>
    <UL TYPE="square">
        <LI>大忌祭</LI>
        <LI>風神祭</LI>
    </UL>
    <UL TYPE="circle">
        <LI>三枝祭</LI>
    </UL>
```

TYPE属性を指定



MacOS版Netscape Navigatorでの表示例（Internet Explorerでも同様）



Windows版Netscape Navigatorでの表示例（Internet Explorerでも同様）

なお、テキストブラウザでは記号を区別して表示することができません。

●COMPACT属性

COMPACT属性を指定することによって、Webブラウザにリストをできるだけ短縮表示するように指示できます。しかし、現実にCOMPACT属性をサポートしているWebブラウザはないようです。

OL

<OL>~</OL>

順序づけリスト（Ordered List）　ブロックレベル要素

内容 ▶ LI
※１個以上

親要素 ▶ BODY
DIV　CENTER　BLOCKQUOTE　FORM　TH　TD　DD
LI

要素の概要

OLは順序づけリストの要素です。順番が重要な意味を持つリストに用います。

開始タグ<OL>と終了タグ</OL>によって、リストの範囲をマークアップします。その中には必ず1個以上のリスト項目「LI要素」（P.71参照）が含まれます。LI要素以外の要素を含むことはできません。

```
<P>詔書の起草から施行まで</P>
<OL>                                    ← 開始タグ
  <LI>内記が御所において草案を作成する</LI>
  <LI>天皇が宸筆で日付を記入し（御画日）、中務卿に給う</LI>
  <LI>中務省ではこれを留めて案とし、写しを作成する</LI>
  <LI>写しに中務卿、大輔、少輔が位署を加える</LI>
  <LI>さらに内印を押して太政官に送る</LI>
  <LI>太政官では大納言以上が位署を加える</LI>
  <LI>さらに施行を詔う文言と日付を記入し、大納言が天皇に覆奏する
  </LI>
  <LI>天皇が許可する場合は、宸筆で「可」と記入する（御画可）</LI>
  <LI>太政官ではこれを留めて案とし、写しを作成して施行する</LI>
</OL>                                   ← 終了タグ
```

リスト項目

詔書の起草から施行まで

1. 内記が御所において草案を作成する
2. 天皇が宸筆で日付を記入し（御画日）、中務卿に給う
3. 中務省ではこれを留めて案とし、写しを作成する
4. 写しに中務卿、大輔、少輔が位署を加える
5. さらに内印を押して太政官に送る
6. 太政官では大納言以上が位署を加える
7. さらに施行を詔う文言と日付を記入し、大納言が天皇に覆奏する
8. 天皇が許可する場合は、宸筆で「可」と記入する（御画可）
9. 太政官ではこれを留めて案とし、写しを作成して施行する

アラビア数字でリストに番号が振られる

OL要素の使用例

なお、LI要素はサンプル画像のようにインデントして表示されますが、この表示方法はHTMLで定義されているわけではありません。したがって、環境やWebブラウザによって見え方が異なる可能性もあります。

属性一覧

●TYPE（初期値：1）

値	働き	H3	N4	I4
1	番号をアラビア数字に設定	○	○	○
a	番号をローマ字小文字に設定	○	○	○
A	番号をローマ字大文字に設定	○	○	○
i	番号をローマ数字小文字に設定	○	○	○
I	番号をローマ数字大文字に設定	○	○	○

属性は省略可

●START（初期値：1）

値	働き	H3	N4	I4
number	開始番号の設定	○	○	○

属性は省略可

●COMPACT（論理型）

値	働き	H3	N4	I4
compact	リストの短縮表示	○	×	×

欄外・属性は省略可



number=数値

属性解説

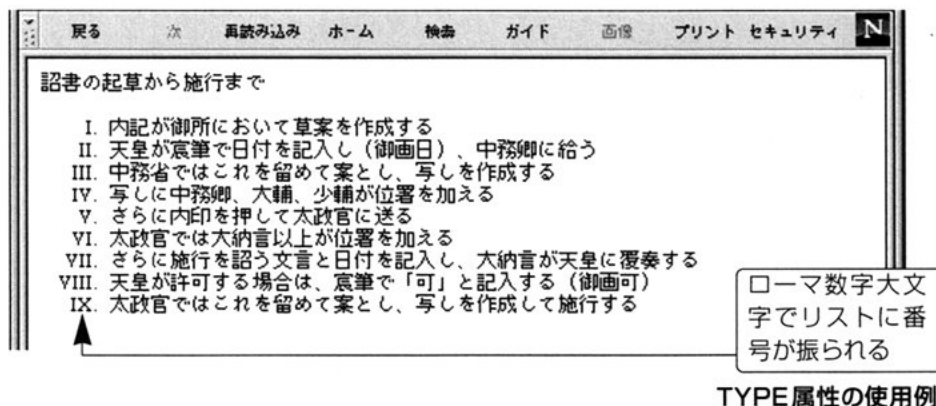
●TYPE属性

Webブラウザが表示する行頭番号の種類を指定します。HTMLでは以下のように表示することを推奨しています。

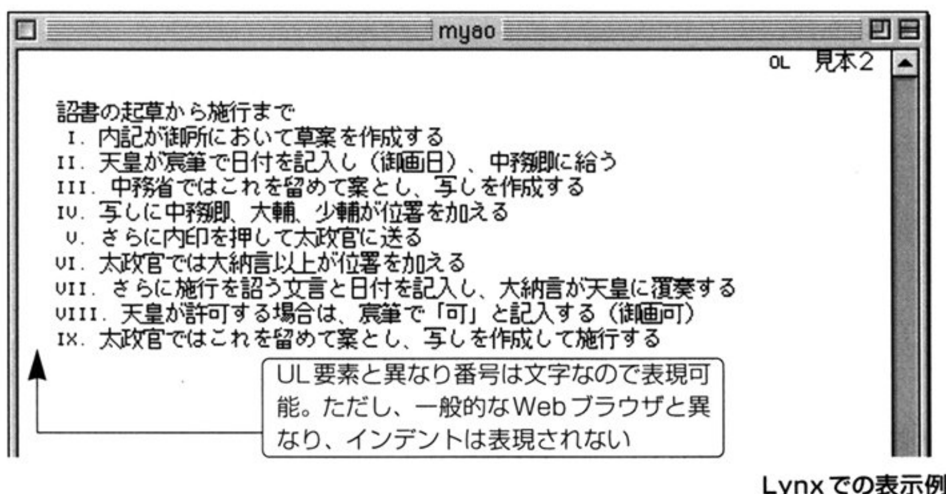
属性値	番号の種類	
1	アラビア数字	1,2,3,4,5,6…
a	ローマ字小文字	a,b,c,d,e,f…
A	ローマ字大文字	A,B,C,D,E,F…
i	ローマ数字小文字	i,ii,iii,iv,v,vi…
I	ローマ数字大文字	I,II,III,IV,V,VI…

```
<P>詔書の起草から施行まで</P>
<OL TYPE="I">
    <LI>内記が御所において草案を作成する</LI>
    <LI>天皇が宸筆で日付を記入し（御画日）、中務卿に給う</LI>
    <LI>中務省ではこれを留めて案とし、写しを作成する</LI>
    <LI>写しに中務卿、大輔、少輔が位署を加える</LI>
    <LI>さらに内印を押して太政官に送る</LI>
    <LI>太政官では大納言以上が位署を加える</LI>
    <LI>さらに施行を詔う文言と日付を記入し、大納言が天皇に覆奏する
</LI>
    <LI>天皇が許可する場合は、宸筆で「可」と記入する（御画可）</LI>
    <LI>太政官ではこれを留めて案とし、写しを作成して施行する</LI>
</OL>
```

TYPE属性で番号をローマ数字大文字に指定



TYPE属性の使用例



Lynxでの表示例

●START属性

順序づけリストを任意の番号から開始したい場合、START属性で開始番号を指

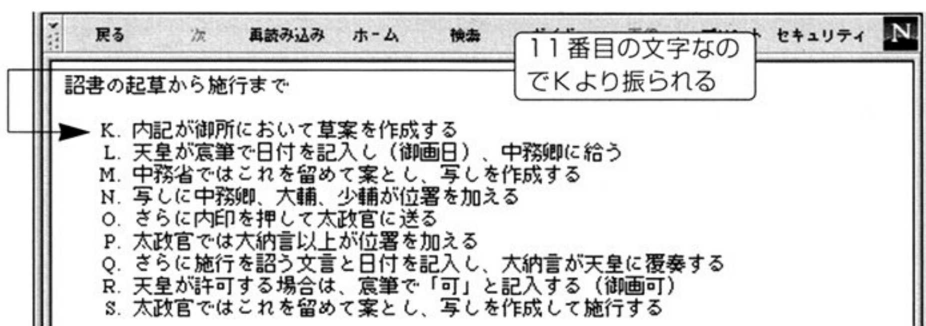
定します。初期値は1です。

設定は全て数値で行いますが、TYPE属性の値によってWebブラウザは以下のように表示します。

TYPE属性の値	Webブラウザの表示
1	5
a	e
A	E
i	v
I	V

```
<P>詔書の起草から施行まで</P>
<OL TYPE="A" START="11">
    <LI>内記が御所において草案を作成する</LI>
    <LI>天皇が宸筆で日付を記入し（御画日）、中務卿に給う</LI>
    <LI>中務省ではこれを留めて案とし、写しを作成する</LI>
    <LI>写しに中務卿、大輔、少輔が位署を加える</LI>
    <LI>さらに内印を押して太政官に送る</LI>
    <LI>太政官では大納言以上が位署を加える</LI>
    <LI>さらに施行を詔う文言と日付を記入し、大納言が天皇に覆奏する
</LI>
    <LI>天皇が許可する場合は、宸筆で「可」と記入する（御画可）</LI>
    <LI>太政官ではこれを留めて案とし、写しを作成して施行する</LI>
</OL>
```

TYPE属性でローマ字大文字
START属性ではじまりを
11番目の文字に指定



START属性の使用例

●COMPACT属性

COMPACT属性を指定することによって、Webブラウザにリストをできるだけ短縮表示するように指示できます。しかし、現実にCOMPACT属性ををサポートしているWebブラウザはないようです。

LI

<LI>~</LI>

リスト項目（List Item）　　ブロックレベル要素

内容 ▶ P UL OL DL (DIR) (MENU) PRE DIV CENTER BLOCKQUOTE FORM ISINDEX HR TABLE TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE INPUT SELECT TEXTAREA A IMG APPLET FONT BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 ▶ UL OL

要素の概要

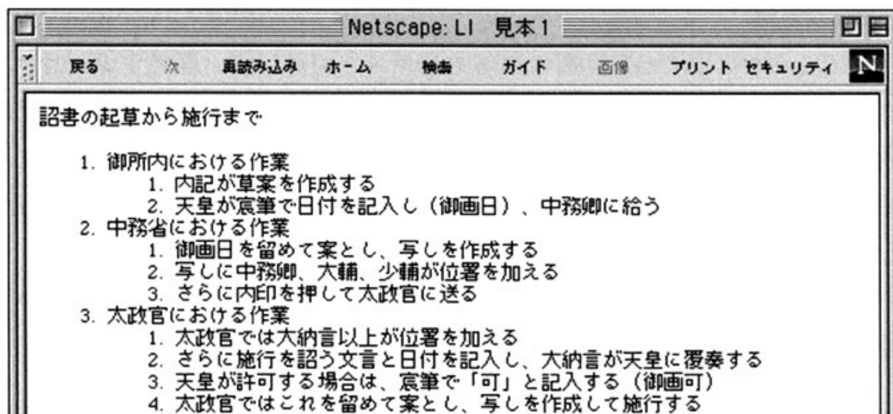
LI要素は、UL要素やOL要素の具体的なリスト項目です。開始タグ<LI>と終了タグ</LI>でリスト項目をマークアップします。終了タグ</LI>は省略可能です。

LI要素の内容はブロックレベル要素とテキストレベル要素、一般的なテキストです。ですから、たとえばLI要素の中にさらにUL要素、OL要素、DL要素などを置いて、入れ子にすることもできます。

なお、LI要素は必ずUL要素とOL要素の内容として使用します。単独で用いたり、他の要素の中で使用することはできません。

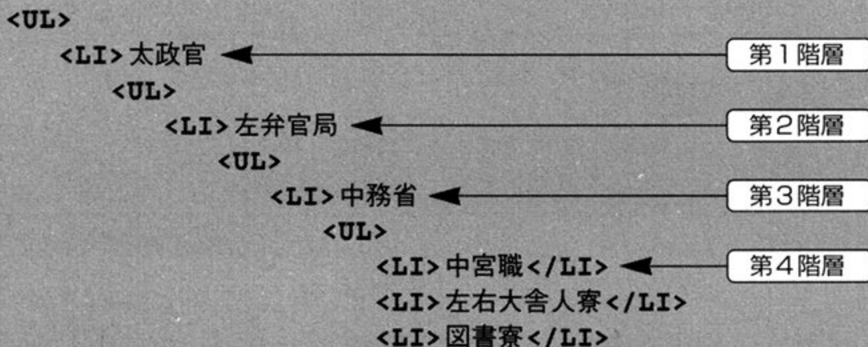
```
<P>詔書の起草から施行まで</P>
<OL>
    <LI>御所内における作業
        <OL>
            <LI>内記が草案を作成する</LI>
            <LI>天皇が宸筆で日付を記入し（御画日）、中務卿に給う
</LI>
        </OL>
    </LI>
    <LI>中務省における作業
        <OL>
            <LI>御画日を留めて案とし、写しを作成する</LI>
            <LI>写しに中務卿、大輔、少輔が位署を加える</LI>
            <LI>さらに内印を押して太政官に送る</LI>
```

```
        </OL>
    </LI>
    <LI>太政官における作業
        <OL>
            <LI>太政官では大納言以上が位署を加える</LI>
            <LI>さらに施行を詔う文言と日付を記入し、大納言が天皇に覆
奏する</LI>
            <LI>天皇が許可する場合は、宸筆で「可」と記入する（御画可）
</LI>
            <LI>太政官ではこれを留めて案とし、写しを作成して施行する
</LI>
        </OL>
    </LI>
</OL>
```

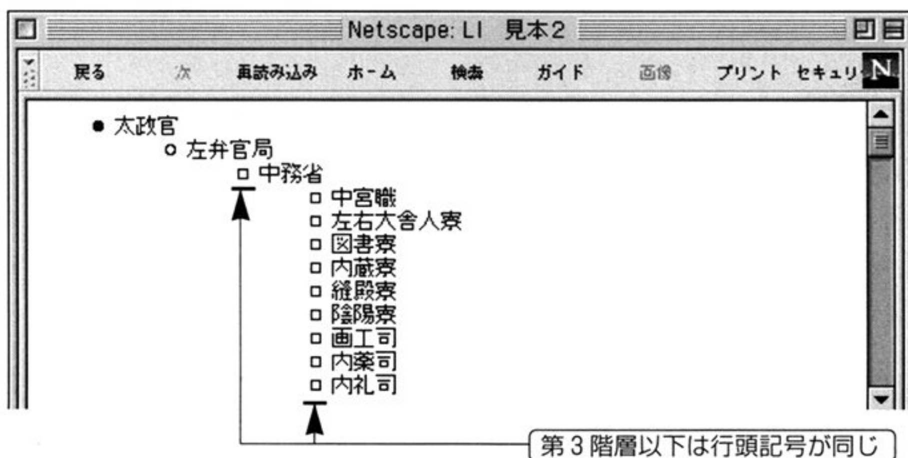


LI要素を入れ子にした表示例

また、UL要素内で入れ子にしTYPE属性を省略した場合、階層に応じて自動的に行頭記号が変化します。ただし、第3階層以下の記号は同一になります。



```
                    <LI>内蔵寮</LI>
                    <LI>縫殿寮</LI>
                    <LI>陰陽寮</LI>
                    <LI>画工司</LI>
                    <LI>内薬司</LI>
                    <LI>内礼司</LI>
                </UL>
            </LI>
        </UL>
        </LI>
    </UL>
  </LI>
</UL>
```

LI要素を階層化した際の表示例

属性一覧

●TYPE（UL内で使用する場合、初期値：disc）

値	働き	H3	N4	I4
disc	行頭記号を●で表示	○	○	○
square	行頭記号を□で表示	○	○	○
circle	行頭記号を○で表示	○	○	○

属性は省略可

●TYPE（OL内で使用する場合、初期値：1）

値	働き	H3	N4	I4
1	番号をアラビア数字に設定	○	○	○
a	番号をローマ字小文字に設定	○	○	○
A	番号をローマ字大文字に設定	○	○	○
i	番号をローマ数字小文字に設定	○	○	○
I	番号をローマ数字大文字に設定	○	○	○

属性は省略可

●VALUE（初期値：1）

値	働き	H3	N4	I4
number	開始番号の設定	○	○	○

属性は省略可

変数 *number*=数値

属性解説

●TYPE属性

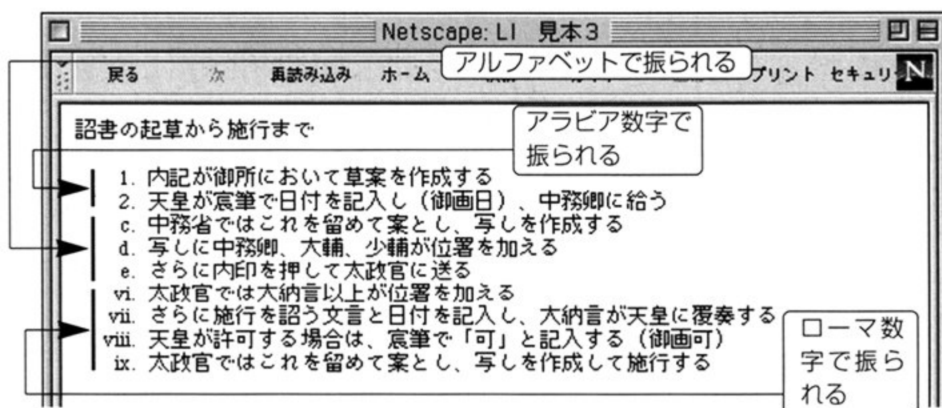
Webブラウザが表示する行頭記号や番号の種類を指定します。種類や指定方法はUL要素やOL要素と同じですが、同じ階層のリストの途中で種類を変更できる点が異なります。

```
<P>詔書の起草から施行まで</P>
<OL>
    <LI TYPE="1">内記が御所において草案を作成する</LI>
    <LI>天皇が宸筆で日付を記入し（御画日）、中務卿に給う</LI>
    <LI TYPE="a">中務省ではこれを留めて案とし、写しを作成する</LI>
    <LI>写しに中務卿、大輔、少輔が位署を加える</LI>
    <LI>さらに内印を押して太政官に送る</LI>
    <LI TYPE="i">太政官では大納言以上が位署を加える</LI>
    <LI>さらに施行を詔う文言と日付を記入し、大納言が天皇に覆奏する</LI>
    <LI>天皇が許可する場合は、宸筆で「可」と記入する（御画可）</LI>
    <LI>太政官ではこれを留めて案とし、写しを作成して施行する</LI>
</OL>
```

番号をアラビア数字に指定

番号をローマ数字小文字に指定

番号をローマ字小文字に指定



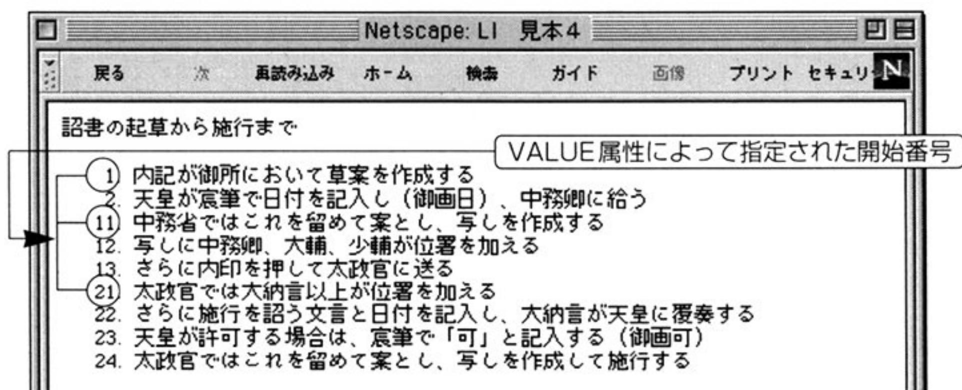
TYPE属性の使用例

●VALUE属性

順序づけリストを任意の番号から開始したい場合、VALUE属性で開始番号を指定します。初期値は1です。基本的な機能や指定方法はOLのSTART属性と同じですが、同じ階層のリストの途中で開始番号を変更できる点が異なります。

```
<P>詔書の起草から施行まで</P>
<OL>
    <LI VALUE="1">内記が御所において草案を作成する</LI>
    <LI>天皇が宸筆で日付を記入し（御画日）、中務卿に給う</LI>
    <LI VALUE="11">中務省ではこれを留めて案とし、写しを作成する
</LI>
    <LI>写しに中務卿、大輔、少輔が位署を加える</LI>
    <LI>さらに内印を押して太政官に送る</LI>
    <LI VALUE="21">太政官では大納言以上が位署を加える
</LI>
    <LI>さらに施行を詔う文言と日付を記入し、大納言が天皇に覆奏する
</LI>
    <LI>天皇が許可する場合は、宸筆で「可」と記入する（御画可）</LI>
    <LI>太政官ではこれを留めて案とし、写しを作成して施行する</LI>
</OL>
```

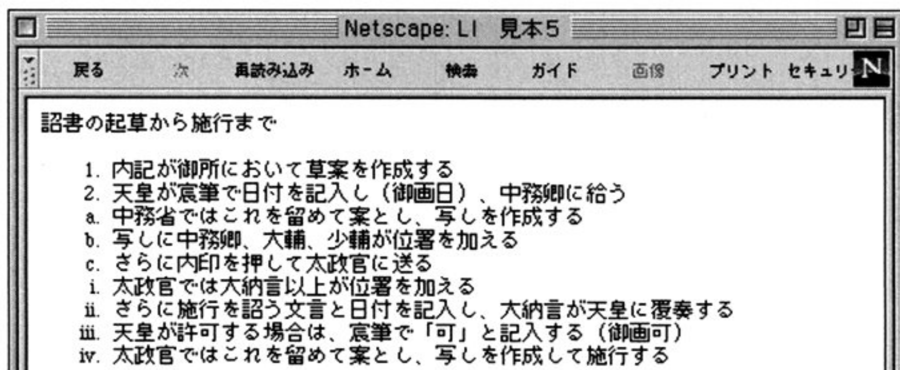
開始番号を1に指定
開始番号を11に指定
開始番号を21に指定



VALUE属性の使用例

また、TYPE属性と組み合わせることもできます。

```
<P>詔書の起草から施行まで</P>
<OL>
    <LI TYPE="1" VALUE="1">内記が御所において草案を作成する
</LI>
    <LI>天皇が宸筆で日付を記入し（御画日）、中務卿に給う</LI>
    <LI TYPE="a" VALUE="1">中務省ではこれを留めて案とし、写しを
作成する</LI>
    <LI>写しに中務卿、大輔、少輔が位署を加える</LI>
    <LI>さらに内印を押して太政官に送る</LI>
    <LI TYPE="i" VALUE="1">太政官では大納言以上が位署を加える
</LI>
    <LI>さらに施行を詔う文言と日付を記入し、大納言が天皇に覆奏する
</LI>
    <LI>天皇が許可する場合は、宸筆で「可」と記入する（御画可）</LI>
    <LI>太政官ではこれを留めて案とし、写しを作成して施行する</LI>
</OL>
```



TYPE属性とVALUE属性を組み合わせた例

DL

<DL>~</DL>

定義型リスト（Definition List） **ブロックレベル要素**

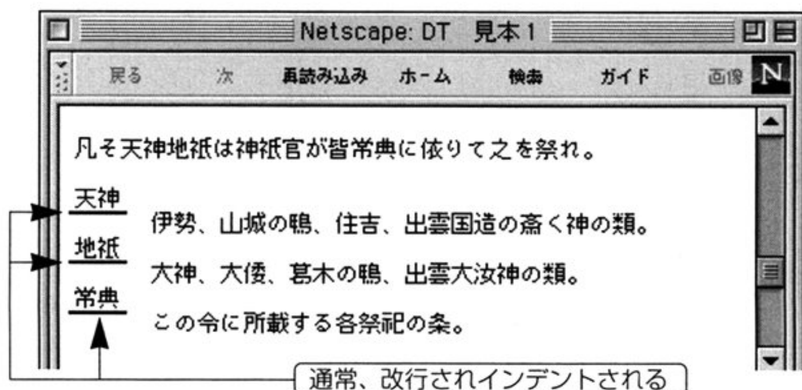
内容 DT DD
※いずれか1個以上

親要素 BODY
DIV CENTER BLOCKQUOTE FORM TH TD DD
LI

要素の概要

DL要素は用語の定義（説明）リストの要素です。開始タグ<DL>と終了タグ</DL>によって、定義リストの範囲をマークアップします。具体的な定義リストの内容は、定義される用語であるDT要素（P.81参照）、その用語の定義であるDD要素（P.82参照）で構成されます。また、このDT要素とDL要素は通常セットで使用されます。

```
<P>凡そ天神地祇は神祇官が皆常典に依りて之を祭れ。</P>
<DL>   ← 開始タグ
 <DT>天神</DT>   ← 用語
  <DD>伊勢、山城の鴨、住吉、出雲国造の斎く神の類。</DD>   ← 定義
 <DT>地祇</DT>   ← 用語
  <DD>大神、大倭、葛木の鴨、出雲大汝神の類。</DD>   ← 定義
 <DT>常典</DT>   ← 用語
  <DD>この令に所載する各祭祀の条。</DD>   ← 定義
</DL>   ← 終了タグ
```



DL要素の使用例

なお、DL要素はサンプル画像のようにDT要素の後ろで改行、DD要素の前でインデントして表示されますが、この表示方法はHTMLで定義されているわけではありません。したがって、環境やWebブラウザによって見え方が異なる可能性もあります。

注意　文法上はDT要素とDD要素のどちらか一方が1個以上あれば、違反にはなりません。また、複数のDT要素に対して1個のDD要素が、反対に1個のDT要素に対して複数のDD要素が存在する（DD要素の連続）こともできます。ちなみにDD要素の連続はHTML 2.0では推奨されていませんが、HTML 3.2では特にその是非には触れられていません。HTML 4.0では仕様書にDD要素の連続使用例が載せてあるので、そのような使用方法も問題ないようです（たとえば、辞書などで1つの用語に対して複数の定義がある場合など）。

属性一覧

●COMPACT（論理型）

値	働き	H3	N4	I4
compact	リストの短縮表示	○	○	○

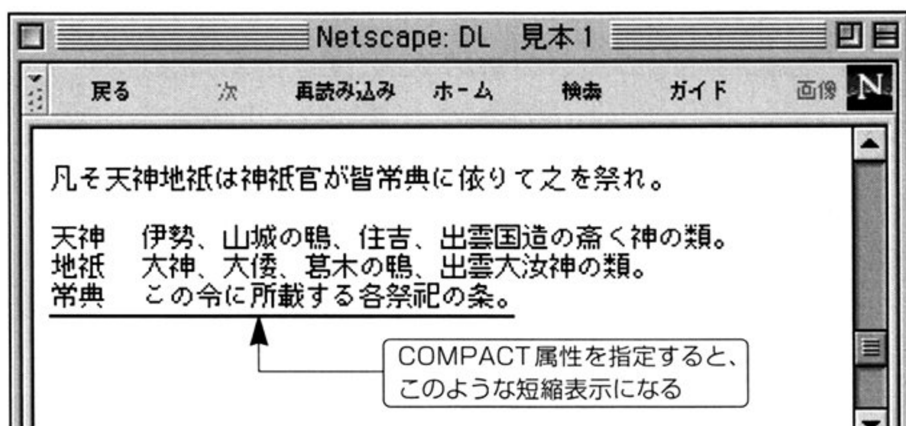
属性は省略可

属性解説

●COMPACT属性

COMPACT属性を指定することによって、Webブラウザにリストをできるだけ短縮表示するよう指示できます。COMPACT属性を指定する場合は、属性名は省略し属性値「compact」のみを記述します(P.8参照)。

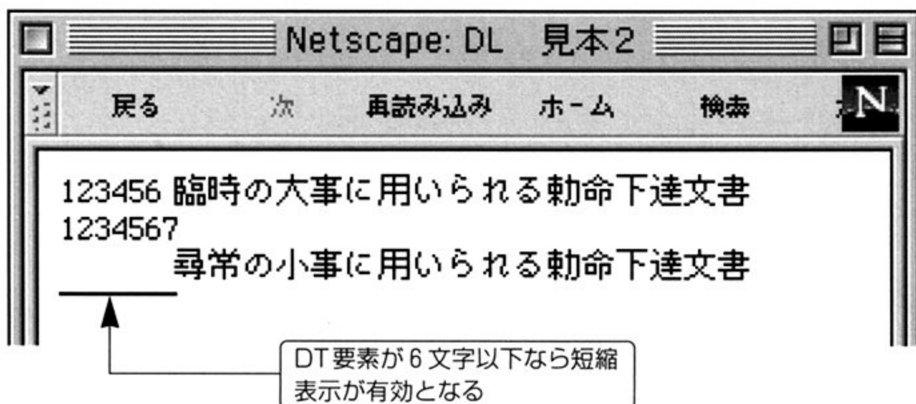
```
<P>凡そ天神地祇は神祇官が皆常典に依りて之を祭れ。</P>
  <DL compact> ◀── COMPACT属性の指定
    <DT>天神</DT>
      <DD>伊勢、山城の鴨、住吉、出雲国造の斎く神の類。</DD>
    <DT>地祇</DT>
      <DD>大神、大倭、葛木の鴨、出雲大汝神の類。</DD>
    <DT>常典</DT>
      <DD>この令に所載する各祭祀の条。</DD>
  </DL>
```



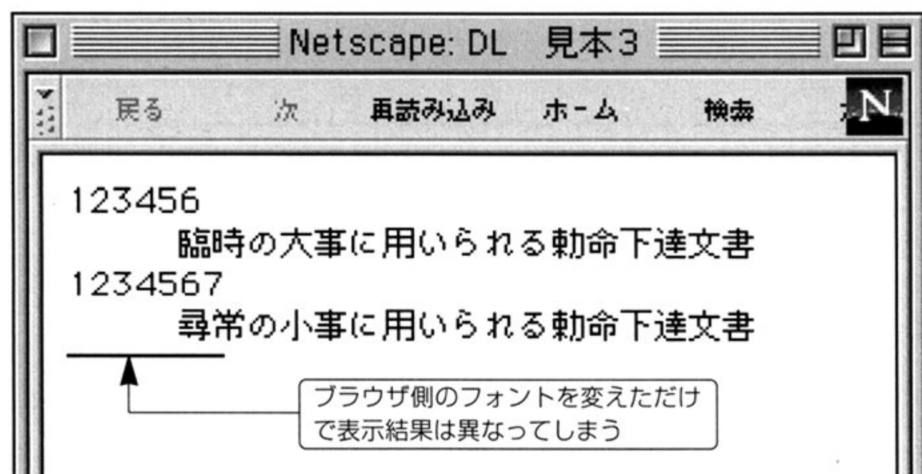
COMPACT属性の使用例

注意　多くのWebブラウザにおいて、COMPACT属性はDT要素の長さがDD要素のインデント幅より短い場合のみ表示に反映されます。しかし、DD要素のインデント幅自体はHTMLに規定がありません。したがって、COMPACT属性の有効性は環境やWebブラウザ、あるいは表示フォントの種類やサイズによって異なります。閲覧者、およびその環境が限定されていない場合は、使用してもあまり意味をなさないこともあるでしょう。

```
<DL compact>
    <DT>123456</DT>
        <DD>尋常の小事に用いられる勅命下達文書</DD>
    <DT>1234567</DT>
        <DD>尋常の小事に用いられる勅命下達文書</DD>
</DL>
```



Mac OS版のNetscape Navigator 4.0で細明朝体・12ポイントを使用した例



表示フォントをOsaka・12ポイントに変更した例

DT

<DT>~</DT>

用語 (Definition Term) ブロックレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR AP INPUT SELECT TEXTAREA
一般的なテキスト

親要素 ▶ DL

要素の概要

DT要素は、DD要素で定義される用語です。開始タグ<DT>と終了タグ</DT>で用語をマークアップします。終了タグ</DT>は、省略可能です。

必ずDL要素の内容として使用されます。また、DT要素の内容はテキストレベル要素と一般的なテキストのみです。

属性一覧

なし

属性解説

なし

DD

<DD>~</DD>

用語の定義（Definition Description）　ブロックレベル要素

内容 P UL OL DL (DIR) (MENU) PRE DIV CENTER BLOCKQUOTE FORM ISINDEX HR TABLE TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE INPUT SELECT TEXTAREA A IMG APPLET FONT BASEFONT BR MAP SCRIPT
一般的なテキスト

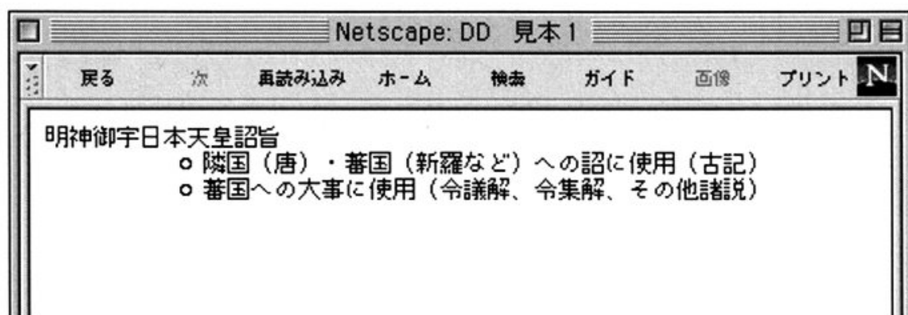
親要素 DL

要素の概要

DD要素は、DT要素の定義（用語の説明）です。開始タグ<DD>と終了タグ</DD>で定義をマークアップします。終了タグ</DD>は省略可能です。

DD要素の内容はブロックレベル要素とテキストレベル要素、一般的なテキストです。ですから、たとえばDD要素の中にさらにUL要素、OL要素、DL要素などを置いて、入れ子にすることもできます。

```
<DL>
 <DT>明神御宇日本天皇詔旨</DT>
  <DD>
  <UL>
    <LI>隣国（唐）・蕃国（新羅など）への詔に使用（古記）</LI>
    <LI>蕃国への大事に使用（令議解、令集解、その他諸説）</LI>
   </UL>
  </DD>
</DL>
```

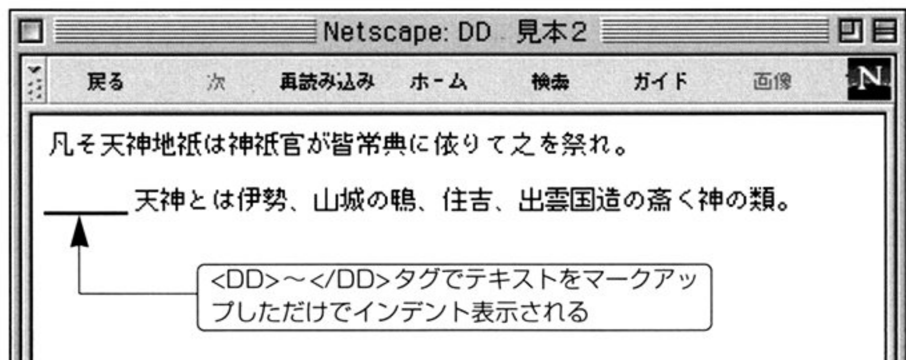


DD要素の使用例

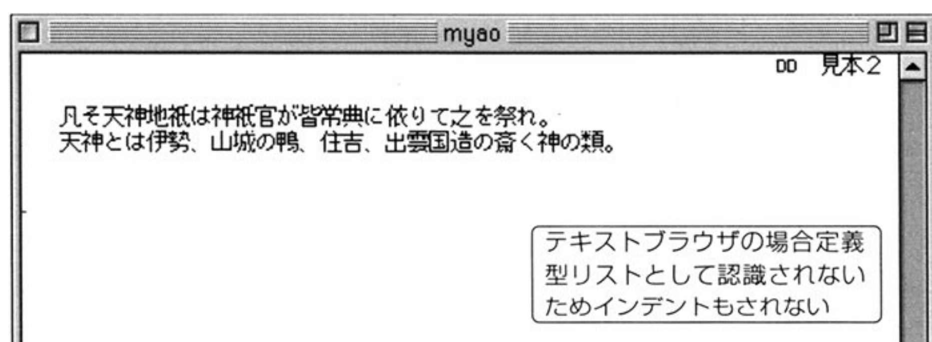
なお、DDは必ずDLの要素として使用されます。単独で用いたり、他の要素の中で使用することはできません。

注意 HTMLにはテキストをインデントさせるための要素はありませんが、文書の内容によってはどうしてもインデントが必要なこともあるかもしれません。一部のHTML文書作成ツールでは、<DD>～</DD>タグでテキストをマークアップしてインデントを表現していますが、Netscape NavigatorやInternet Explorerではインデント表示されても、他のWebブラウザでもそうなるとは限りません。たとえば、LynxではDL・DT・DDの3要素がきちんと揃っていれば、それらは「定義型リスト要素」として認識されDD要素はインデント表示されますが、単に<DD>～</DD>タグでマークアップしただけのテキストはインデントされません。また、Netscape NavigatorやInternet Explorerでも将来要素の解釈が厳密になった場合、インデントされなくなる可能性もあります。

```
<P>凡そ天神地祇は神祇官が皆常典に依りて之を祭れ。</P>
<DD>天神とは伊勢、山城の鴨、住吉、出雲国造の斎く神の類。</DD>
```



DD要素の使用例



Lynxでの表示例

どうしてもインデントが必要であれば、単に<DD>～</DD>タグを使用するのではなく、DL・DT・DDの3要素を揃えて見かけ上きちんと「定義型リスト要素」の体裁をとっておく方がいいかもしれません。

属性一覧

なし

属性解説

なし

PRE

<PRE>~</PRE>

整形済みテキスト（PREformatted text）　ブロックレベル要素

内容 ▶ TT I B U STRIKE EM STRONG DFN CODE SAMP KBD VAR CITE INPUT SELECT TEXTAREA A APPLET BASEFONT BR MAP SCRIPT 一般的なテキスト

親要素 ▶ BODY DIV CENTER BLOCKQUOTE FORM TH TD DD LI

要素の概要

PRE要素は、内容が「整形済みテキスト」であることを示す要素です。Webブラウザで表示される際に空白文字や改行コード、タブが見た目のまま有効になりますので、簡単な表やどうしても固定したレイアウトを維持しなければならない場合（たとえば電子メールの本文をそのまま利用する場合など）に使用できます。また、レイアウトを維持するために、画面表示には等幅フォントが使用されます。

開始タグ<PRE>と終了タグ</PRE>によって整形されたテキストをマークアップします。その中には0個以上のテキストレベル要素か、一般的なテキストが含まれます。ただし、テキストに変換できない画像やフォントサイズを変更する恐れのある要素は内容として使用できません。

なお、同様の機能を持つXMP・LISTING・PLAINTEXT要素がありますが、これらはPRE要素が策定される以前の名残です。HTML 3.2では、新規に作成するHTML文書では使用を避けるよう求められていますので、リファレンスの内容一覧には載せていません（HTML 4.0では完全に廃止されている）。ただし、Webブラウザによっては古い文書との互換性をとるために、これらの要素を認識できる場合もあります。

```
<PRE>  ← 開始タグ

家令職員令

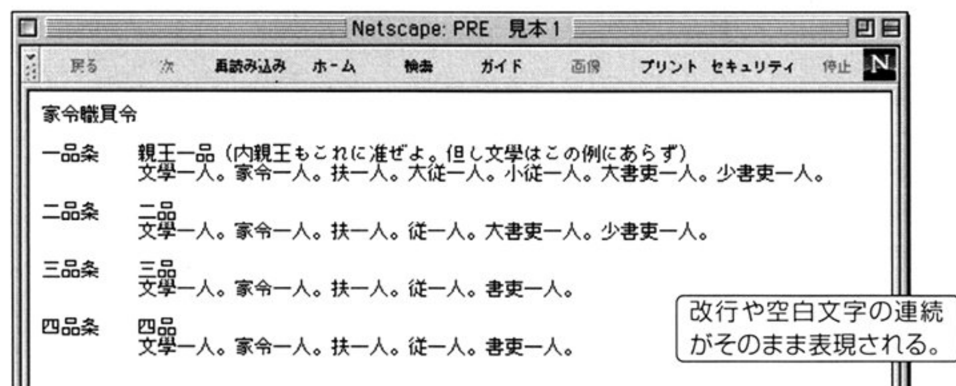
一品条　　親王一品（内親王もこれに准ぜよ。但し文學はこの例にあらず）
　　　　　文學一人。家令一人。扶一人。大従一人。小従一人。大書吏一人。
少書吏一人。
```

```
二品条　　二品
　　　　　文學一人。家令一人。扶一人。従一人。大書吏一人。少書吏一人。

三品条　　三品
　　　　　文學一人。家令一人。扶一人。従一人。書吏一人。

四品条　　四品
　　　　　文學一人。家令一人。扶一人。従一人。書吏一人。
</PRE>
```

終了タグ



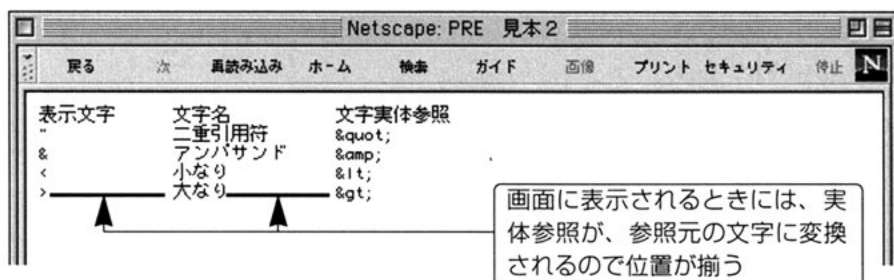
PRE要素の使用例

PRE要素ではタブによるレイアウトも可能ですが、HTML文書作成ツールによっては、タブを適当な数の通常の空白文字に置き換えて整形するものがあります。そのため、作者の意図したレイアウトが崩れてしまうこともありますので、一般的にタブの使用は推奨されていません。少々面倒ですが、PRE要素にするテキストでは最初からタブの代わりに空白文字を使用しましょう。

なお、何行にもわたるテキストをPRE要素にした場合、改行位置によっては閲覧者が読みにくくなる場合があります。たとえば、モニタ画面の解像度が低い場合（あるいは表示フォントを大きくしている場合）、多くのWebブラウザではウインドウの端で自動折り返しされないため、横スクロールしなければテキストを読むことができません。かといって、早めに改行する（行を短く整形する）と、モニタ画面の解像度が高いユーザには間抜けなレイアウトに見えます。また、日本語フォントだけのテキストの場合はあまり気にならないかもしれませんが、大量の欧文テキストを等幅フォントで表示すると、美しくない上に非常に読みにくく感じる人もいます。

注意 「<」「>」「"」など特殊な文字は、マークアップに使用される記号と同じなので、そのままHTML文書内に記述しても表示することができません。この原則はPRE要素においても同様なので、表示する場合は文字参照（P.310参照）という方法を使用します。
PRE要素で文字参照を使用する場合、その幅に注意します。文字参照の記述が数文字になっても、実際に表示されるのは1文字分です。

```
<PRE>
表示文字         文字名              文字実体参照
&quot;              二重引用符          &amp;quot;
&amp;              アンパサンド          &amp;amp;
&lt;                小なり             &amp;lt;
&gt;                大なり             &amp;gt;
</PRE>
```



属性一覧

●WIDTH（初期値：80）

値	働き	H3	N4	I4
number	表示幅の指定	○	×	×

属性は省略可

変数 *number*=数値

属性解説

●WIDTH属性

WIDTH属性を指定することによって、Webブラウザに表示幅を指示できます。しかし、現実にWIDTH属性をサポートしているWebブラウザはないようです。

DIV

<DIV>~</DIV>

要素のグループ化と位置揃え（document DIVisions）　ブロックレベル要素

内容 H1 H2 H3 H4 H5 H6
ADDRESS
P UL OL DL (DIR) (MENU) PRE DIV CENTER
BLOCKQUOTE FORM ISINDEX HR TABLE
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 BODY
DIV CENTER BLOCKQUOTE FORM TH TD DD
LI

要素の概要

Webブラウザで要素を位置揃えして表示するために、いくつかの要素にはALIGN属性が用意されています。たとえば、P要素でALIGN属性を使用すれば、1つの段落を丸ごと整列することができます。では、複数のP要素を位置揃えする場合は、個々のP要素にALIGN属性を指定しなければならないのでしょうか？いえ、そんなときはDIV要素を使用するといいでしょう。

DIV要素は、複数のブロックレベル要素・テキストレベル要素・一般的なテキストをグループ化し、まとめて位置揃えする場合に使用されます。開始タグ<DIV>と終了タグ</DIV>によってマークアップされた範囲は、DIV要素のALIGN属性に従って位置揃えされます。

属性一覧

●ALIGN

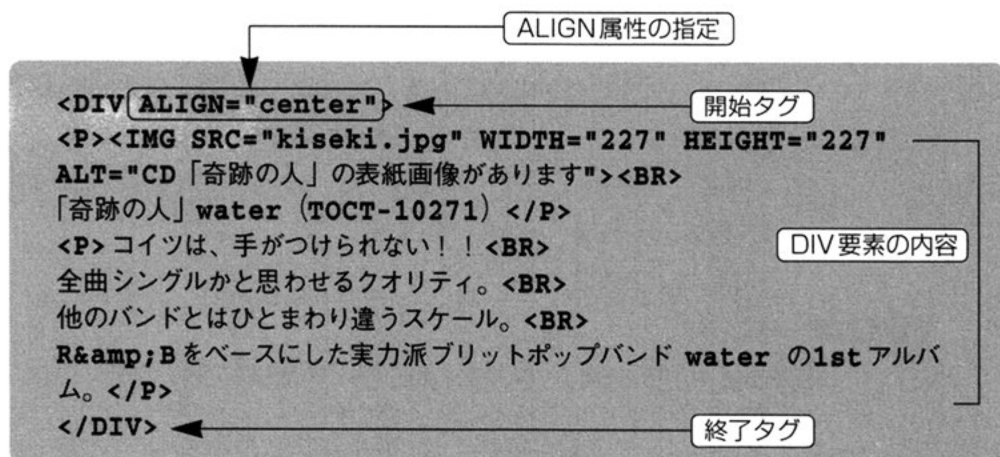
値	働き	H3	N4	I4
left	左揃え	○	○	○
center	中央揃え	○	○	○
right	右揃え	○	○	○

属性は省略可

属性解説

●ALIGN属性

DIV要素の位置揃えを指定します。属性値にはleft、center、rightの3種類があります。属性を省略した場合は、一般的にleftが適用されます。




```
<DIV ALIGN="center">
<P><IMG SRC="kiseki.jpg" WIDTH="227" HEIGHT="227"
ALT="CD「奇跡の人」の表紙画像があります"><BR>
「奇跡の人」water（TOCT-10271）</P>
<P>コイツは、手がつけられない！！<BR>
全曲シングルかと思わせるクオリティ。<BR>
他のバンドとはひとまわり違うスケール。<BR>
R&amp;Bをベースにした実力派ブリットポップバンド water の1st アルバム。</P>
</DIV>
```



DIV要素の使用例

NN 3 | IE 3 | NN 4 | IE 4

CENTER

<CENTER>～</CENTER>

要素のグループ化と中央揃え（text alignment）　ブロックレベル要素

内容 ▶ H1　H2　H3　H4　H5　H6
ADDRESS
P　UL　OL　DL　(DIR)　(MENU)　PRE　DIV　CENTER
BLOCKQUOTE　FORM　ISINDEX　HR　TABLE
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE
INPUT　SELECT　TEXTAREA　A　IMG　APPLET　FONT
BASEFONT BR　MAP　SCRIPT
一般的なテキスト

親要素 ▶ BODY
DIV　CENTER　BLOCKQUOTE　FORM　TH　TD　DD
LI

要素の概要

CENTER要素は、HTML文書内の特定の要素を他から区分し、中央揃えするための要素です。Netscape社の独自拡張ですが、あまりにも普及したために仕方なくHTML 3.2に採用されたものです。

機能的にはDIV要素のALIGN属性にcenterを指定した場合と同じで、開始タグ<CENTER>と終了タグ</CENTER>によってマークアップされた範囲が中央揃えされます。

属性一覧

なし

属性解説

なし

BLOCKQUOTE <BLOCKQUOTE>~<BLOCKQUOTE>

引用文（quoted passage） ブロックレベル要素

内容 H1 H2 H3 H4 H5 H6
ADDRESS
P UL OL DL (DIR) (MENU) PRE DIV CENTER
BLOCKQUOTE FORM ISINDEX HR TABLE
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 BODY
DIV CENTER BLOCKQUOTE FORM TH TD DD
LI

要素の概要

BLOCKQUOTE要素は、他の文書から引用した文書（テキストだけでなく画像なども含む）を明示するための要素です。開始タグ<BLOCKQUOTE>と終了タグ</BLOCKQUOTE>によって引用文をマークアップします。

内容にはテキストレベル要素や一般的なテキストだけでなく、ブロックレベル要素を含むこともできます。あまり使用する機会はないとは思いますが、BLOCKQUOTE要素を入れ子にして「引用文内の引用文」を表すこともできます。

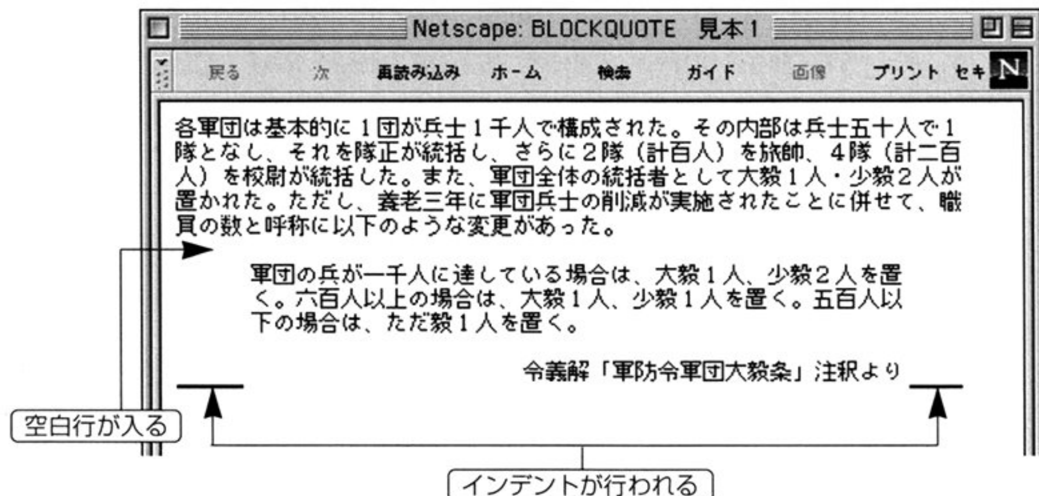
多くのWebブラウザでは、BLOCKQUOTE要素を表示する際に本文と区別する（1つのまとまりとして取り扱う）ために前後に空白行を挿入し、インデントを行います。しかし、これはあくまでもWebブラウザの表示機能によるものですから、BLOCKQUOTE要素を空白行の挿入やインデントの目的で使用してはいけません。

```
<P>各軍団は基本的に1団が兵士1千人で構成された。その内部は兵士五十人
で1隊となし、それを隊正が統括し、さらに2隊（計百人）を旅帥、4隊（計
二百人）を校尉が統括した。また、軍団全体の統括者として大毅1人・少毅2
人が置かれた。ただし、養老三年に軍団兵士の削減が実施されたことに併せて、
職員の数と呼称に以下のような変更があった。</P>
```

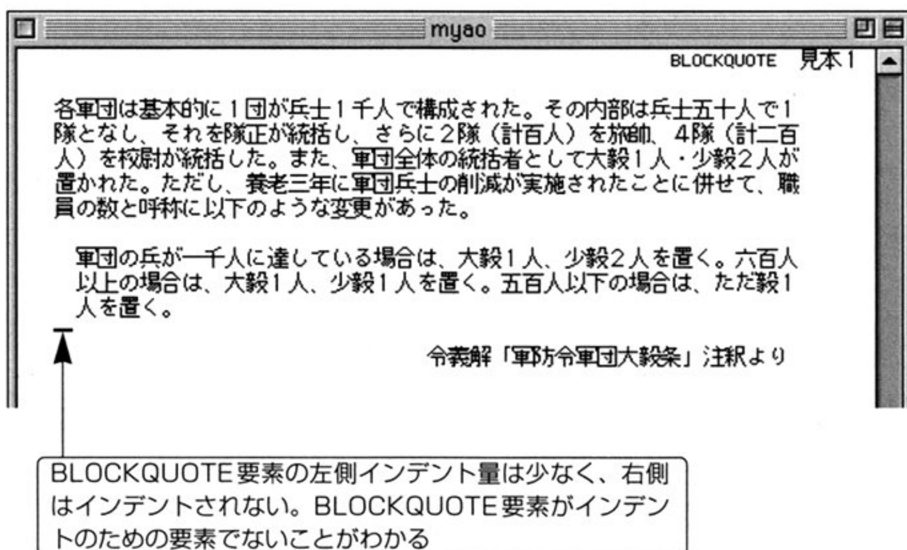
```
<BLOCKQUOTE>
<P>軍団の兵が一千人に達している場合は、大毅1人、少毅2人を置く。六百人以上の場合は、大毅1人、少毅1人を置く。五百人以下の場合は、ただ毅1人を置く。</P>
<P ALIGN="right">令義解「軍防令軍団大毅条」注釈より</P>
</BLOCKQUOTE>
```

開始タグ

終了タグ



BLOCKQUOTE要素の使用例



Lynxでの表示例

HR

<HR>

水平線（Horizontal Rule）　ブロックレベル要素

内 容 ▶ 空

親要素 ▶ BODY
DIV　CENTER　BLOCKQUOTE　FORM　TH　TD　DD
LI

要素の概要

HR要素は、1つの文書内で話題が変わったり意味上の変化がある場合、その区切りを明示するために使用します。Webブラウザではその名の通り水平線が表示されます。また、音声読み上げ機能を持つWebブラウザや専用アプリケーションなどでは、HR要素に遭遇すると読み上げに間を置いて文書の区切りがわかるようにします。

HR要素は、開始タグ<HR>のみでマークアップされるいわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。また、ブロックレベル要素ですから、P要素の中やテキストレベル要素の中に置くことはできません。

属性一覧

●ALIGN（初期値：center）

値	働き	H3	N4	I4
left	左揃え	○	○	○
center	中央揃え	○	○	○
right	右揃え	○	○	○

属性は省略可

●NOSHADE（論理型）

値	働き	H3	N4	I4
noshade	線を平面的にする	○	○	○

属性は省略可

●SIZE

値	働き	H3	N4	I4
pixels	線の太さ（pixel）	○	○	○

属性は省略可

●WIDTH（初期値：100%）

値	働き	H3	N4	I4
length	線の長さ	○	○	○

属性は省略可

変数 *pixels*=pixel数 *length*=pixel数または%

属性解説

●ALIGN属性

水平線を左右の余白内のどの位置に表示するかを指定します。多くの場合、長さを指定するWIDTH属性とセットで使用します(WIDTH属性の項参照)。

●NOSHADE属性

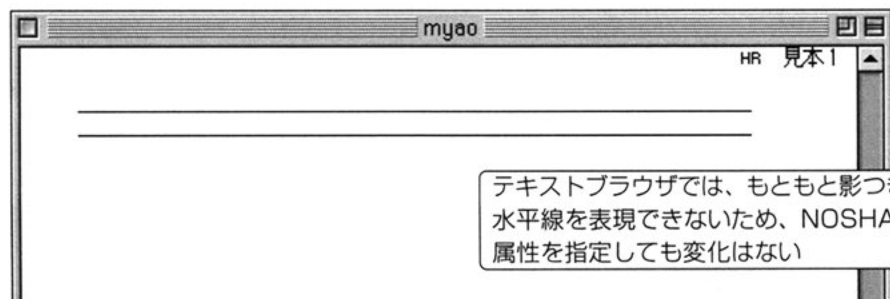
NOSHADE属性を指定することによって、Webブラウザに一般的な陰影つき水平線ではなく、平面的な水平線を引くよう指示します。NOSHADE属性を指定する場合は、属性名は省略し属性値「noshade」のみを記述します(P.8参照)。

```
<HR>
<HR noshade>
```

NOSHADE属性の指定



HR要素の表示例



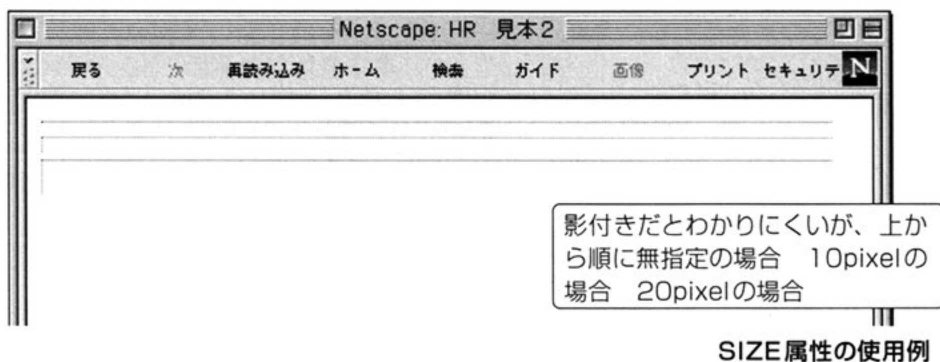
Lynxでの表示例

●SIZE属性

SIZE属性を指定することによって、Webブラウザに水平線を表示する際の太さをpixel単位で指示できます。

また、SIZE属性を省略した場合、Webブラウザなどアプリケーション側で適当な属性値が与えられますが、通常は2pixelが多いようです。ただし、Lynxなどテキストブラウザでは太さを表現することはできません。

```
<HR>
<HR SIZE="10">
<HR SIZE="20">
```



SIZE属性の使用例

●WIDTH属性

WIDTH属性を指定することによって、Webブラウザに水平線を表示する際の長さをWebブラウザのウインドウ幅に対する%単位、または絶対値をpixel単位で指示することができます。また、ALIGN属性とセットで用いると、水平線の表示位置を指定することもできます。

なお、WIDTH属性を省略した場合、Webブラウザ側で適当な属性値が与えられますが、通常は100%が多いようです。ただし、WIDTH="100%"を指定しても完全にウインドウ幅の100%に表示するWebブラウザはないようです。

```
<HR>
<HR WIDTH="80">
<HR WIDTH="80%">
<HR WIDTH="40%">
<HR ALIGN="left" WIDTH="40%">
<HR ALIGN="right" WIDTH="40%">
```



WIDTH属性の使用例

TABLE

<TABLE>～</TABLE>

表（The TABLE element） ブロックレベル要素

内容 ▶ CAPTION TR
※１個以上

親要素 ▶ BODY
DD LI DIV CENTER
BLOCKQUOTE FORM TH TD

要素の概要

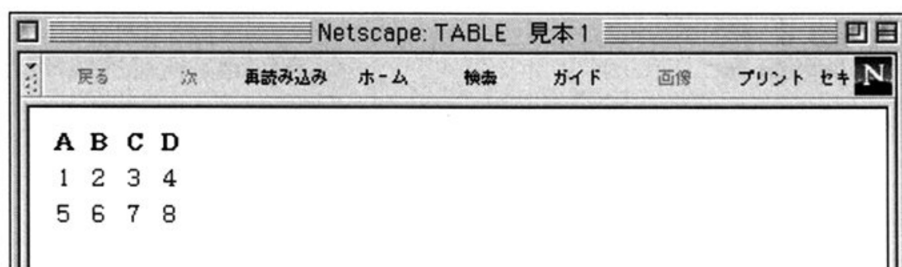
TABLE要素はHTML文書で表組を行う要素です。開始タグ<TABLE>と終了タグ</TABLE>によって、表を構成する各種の要素をマークアップします。

注意 表機能の正式サポートは、HTML 3.2の大きな特徴の１つです。それまでのPRE要素を使用した見せかけの表とは異なり、文書中のあるデータの固まりが「表要素」としての意味を持つことが可能になった意義は大きいでしょう。とはいえ、表を要素として認識はできても、画面上でそれらしく表現できないWebブラウザも存在します。たとえば、Lynxなどテキストブラウザでは表のレイアウトは表現できません（場合によっては、別途PRE要素で整形した表を用意するといいかもしれません）。
一方、表機能をWebブラウザでの表示レイアウトのために応用することもできます。たとえば、文書全体を表要素とし、その中でレイアウトを行っている人も多くいます。しかし、HTML 3.2の仕様書の中でも触れられていますが、表機能をレイアウトのために使用すると、テキストブラウザや音声出力機能を利用しているユーザに正しく情報が伝わらなくなる恐れがあります。
また、多くのWebブラウザでは、表要素は内容が全部受信され終わらないと表示が始まりません。「大きな表」だといつまでたっても画面が空白のままで、何かエラーでも起こっているのではないか？　と心配になってしまいます。くれぐれも、表機能のレイアウトへの応用は慎重に行ってください。

行要素

```
<TABLE>   ← 開始タグ
  <TR><TH> A </TH><TH> B </TH><TH> C </TH><TH> D </TH></TR>
  <TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD><TD> 4 </TD></TR>
  <TR><TD> 5 </TD><TD> 6 </TD><TD> 7 </TD><TD> 8 </TD></TR>
</TABLE>  ← 終了タグ
```

表



表の基本構造

なお、TABLE要素はブロックレベル要素ですが、内容としては表題のCAPTION要素を0個以上、表の行要素であるTR要素を1個(＝1行分)以上置くことができます。それ以外の要素を置くことはできません。

属性一覧

●ALIGN（初期値：left）

値	働き	H3	N4	I4
left	左揃え	○	○	○
center	中央揃え	○	○	○
right	右揃え	○	○	○

属性は省略可

●WIDTH

値	働き	H3	N4	I4
length	表の横幅	○	○	○

属性は省略可

●BORDER（初期値：0）

値	働き	H3	N4	I4
pixels	表の外枠幅	○	○	○

属性は省略可

●CELLSPACING

値	働き	H3	N4	I4
pixels	セルの枠幅	○	○	○

属性は省略可

●CELLPADDING

値	働き	H3	N4	I4
pixels	セルと内容との間隔	○	○	○

属性は省略可

●BORDERCOLOR

値	働き	H3	N4	I4
color	表の枠色	−	−	!

属性は省略可

●BORDERCOLORLIGHT

値	働き	H3	N4	I4
color	表の枠色	−	−	!

属性は省略可

●BORDERCOLORDARK

値	働き	H3	N4	I4
color	表の枠のシャドー色	−	−	!

属性は省略可

変数　*color*=色指定の16進数（#rrggbb）または色名　*pixels*=pixel数
length=pixel数または%

属性解説

●ALIGN属性

表の位置揃えを指定します

ALIGN属性を省略した場合、Webブラウザなどアプリケーション側でleftが与えられます。

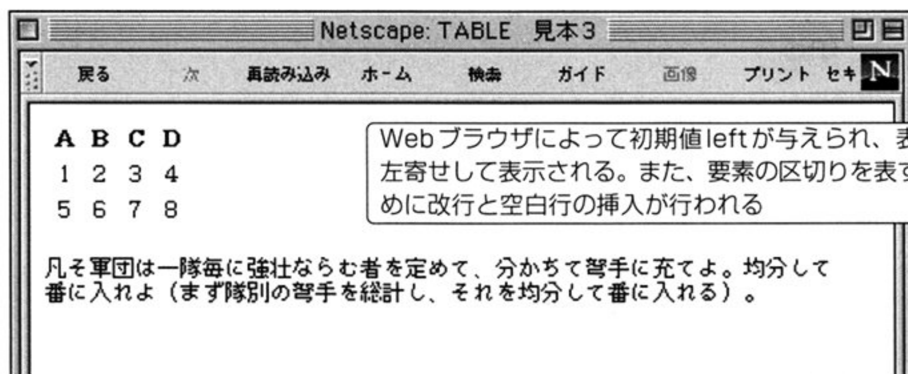
```
<TABLE ALIGN="right">      ← ALIGN属性の指定
<TR><TH> A </TH><TH> B </TH><TH> C </TH><TH> D </TH></TR>
<TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD><TD> 4 </TD></TR>
<TR><TD> 5 </TD><TD> 6 </TD><TD> 7 </TD><TD> 8 </TD></TR>
</TABLE>
```



ALIGN属性をrightに設定した例

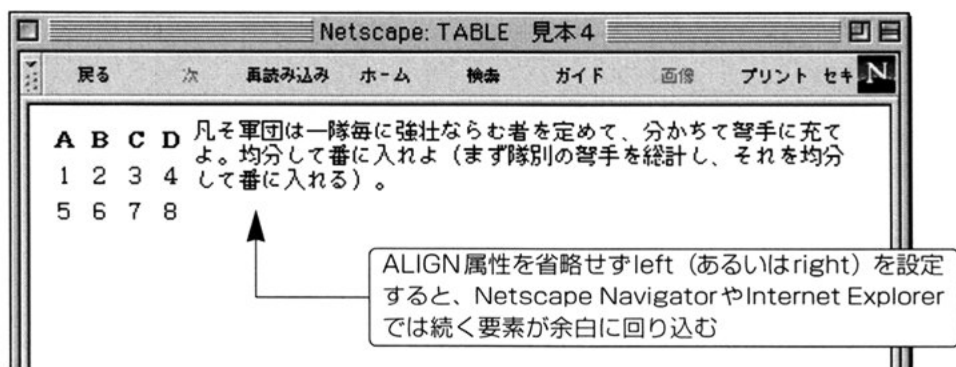
ALIGN属性の使用で注意しなければならないのは、属性値にleftやrightを設定した場合、Webブラウザによっては、TABLE要素に続く他の要素（テキストなど）が余白に回り込んでしまうことです。属性を省略したり、centerを設定した場合はその限りではありません。

```
<TABLE>
<TR><TH> A </TH><TH> B </TH><TH> C </TH><TH> D </TH></TR>
<TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD><TD> 4 </TD></TR>
<TR><TD> 5 </TD><TD> 6 </TD><TD> 7 </TD><TD> 8 </TD></TR>
</TABLE>
<P>凡そ軍団は一隊毎に強壮ならむ者を定めて、分かちて弩手に充てよ。均分
して番に入れよ（まず隊別の弩手を総計し、それを均分して番に入れる）。</P>
```

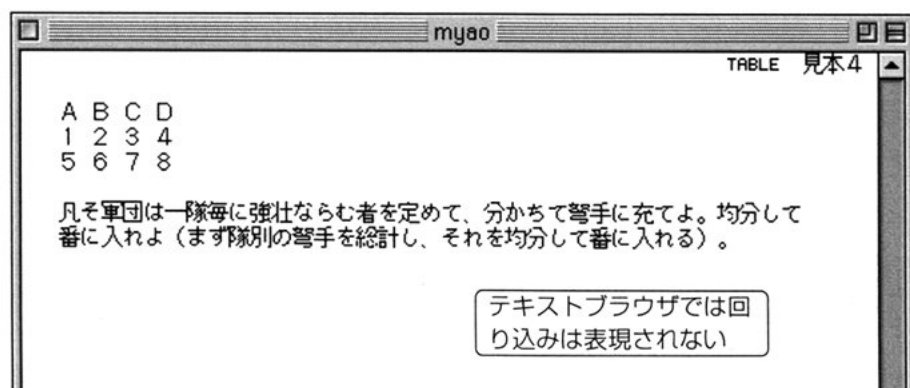


ALIGN属性を省略した例

```
<TABLE ALIGN="left">
(以下略)
```



ALIGN属性にleftを設定した例

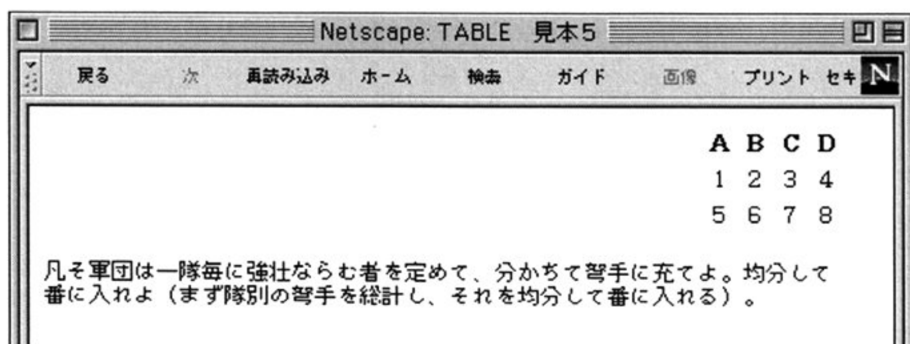


Lynxでの表示例

もし、属性値にrightを設定し、かつ表にテキストが回り込まないようにするには、TABLE要素で位置揃えを指定せず、DIV要素を使用するといいでしょう（TABLE要素では親要素であるDIV要素の属性がそのまま継承される）。

```
<DIV ALIGN="right">  ← DIV要素にALIGN="right"を設定
<TABLE>
<TR><TH> A </TH><TH> B </TH><TH> C </TH><TH> D </TH></TR>
<TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD><TD> 4 </TD></TR>
<TR><TD> 5 </TD><TD> 6 </TD><TD> 7 </TD><TD> 8 </TD></TR>
</TABLE>
</DIV>
<P>凡そ軍団は一隊毎に強壮ならむ者を定めて、分かちて弩手に充てよ。均分
して番に入れよ（まず隊別の弩手を総計し、それを均分して番に入れる）。
</P>
```

表



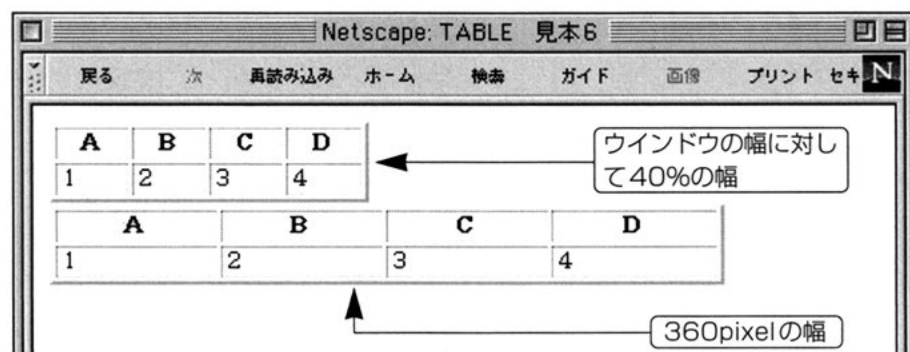
DIV要素にALIGN属性を使用することで回り込みを回避できる

●WIDTH属性

WIDTH属性を指定することによって、Webブラウザに表を表示する際の幅をWebブラウザのウインドウ幅に対する%単位、または絶対値をpixel単位で指定することができます。

WIDTH属性を省略した場合、Webブラウザなどアプリケーション側で適当な属性値が設定されます。また、Lynxなどのテキストブラウザには WIDTH属性を表現できないものがあります。

```
<TABLE BORDER="2" WIDTH="40%">   ← WIDTH属性の指定
<TR><TH> A </TH><TH> B </TH><TH> C </TH><TH> D </TH></TR>
<TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD><TD> 4 </TD></TR>
</TABLE>
<TABLE BORDER="2" WIDTH="360">   ← WIDTH属性の指定
<TR><TH> A </TH><TH> B </TH><TH> C </TH><TH> D </TH></TR>
<TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD><TD> 4 </TD></TR>
</TABLE>
```



WIDTH属性の使用例

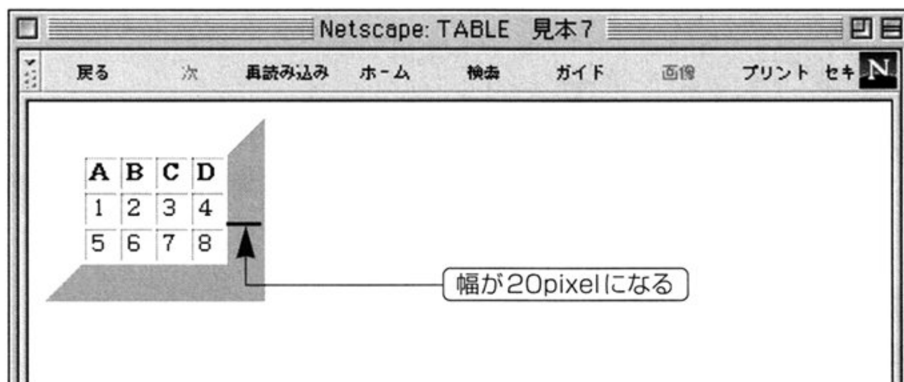
なお、WIDTH属性はできるだけ絶対値で指定しないことをおすすめします。なぜなら、表の幅が固定されると、画面の狭い端末（コンピュータとは限りません）を使用しているユーザは、ウインドウに表が収まらず、表を見渡すために横スクロールを強いられるからです（Apple社のCyberdogのように強引に変形して収めるものもありますが、それは少数派です）。

●BORDER属性

表の外枠の幅をpixel単位で指定します。

BORDER属性を省略した場合、Webブラウザなどアプリケーション側で適当な値が設定されます。一般的にはBORDER="0"が設定されることが多く、その場合は外枠は表示されません。また、Lynxなどのテキストブラウザには枠が表現できないものがあり、その場合はBORDER属性も表現できません。

```
<TABLE BORDER="20">  ← BORDER属性の指定
<TR><TH> A </TH><TH> B </TH><TH> C </TH><TH> D </TH></TR>
<TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD><TD> 4 </TD></TR>
<TR><TD> 5 </TD><TD> 6 </TD><TD> 7 </TD><TD> 8 </TD></TR>
</TABLE>
```

BORDER属性の使用例

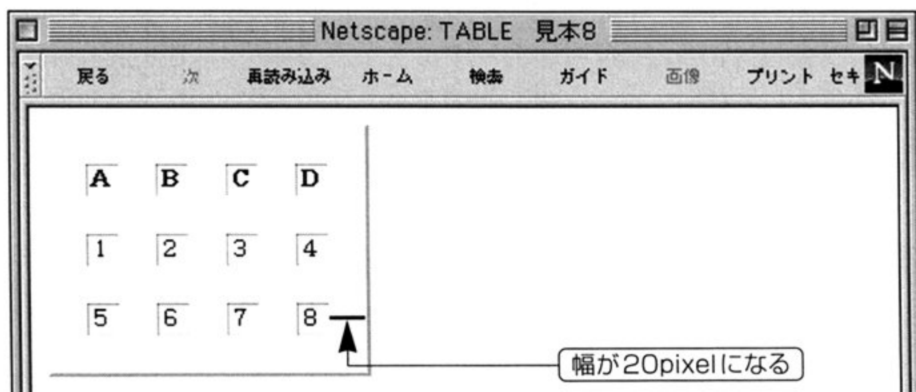
●CELLSPACING属性

表の枠の幅をpixel単位で指定します。

CELLSPACING属性を省略した場合、Webブラウザなどアプリケーション側で適当な値が設定されます。一般的にはCELLSPACING="2"が設定されることが多いようですが、BORDER="0"の場合は、それに合わせてCELLSPACING="0"になります。また、Lynxなどのテキストブラウザには枠が表現できないものがあり、その場合はCELLSPACING属性も表現できません。

CELLSPACING属性の指定

```
<TABLE BORDER="2" CELLSPACING="20">
<TR><TH>A</TH><TH>B</TH><TH>C</TH><TH>D</TH></TR>
<TR><TD>1</TD><TD>2</TD><TD>3</TD><TD>4</TD></TR>
<TR><TD>5</TD><TD>6</TD><TD>7</TD><TD>8</TD></TR>
</TABLE>
```



CELLSPACING属性の使用例

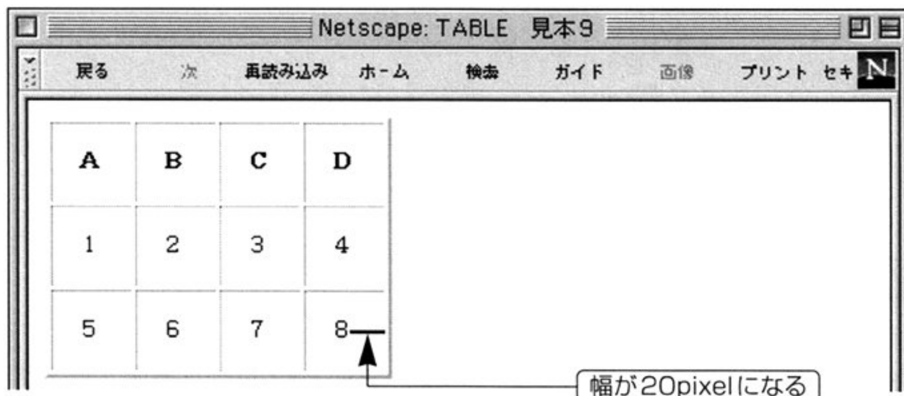
●CELLPADDING属性

表の枠と各セル内のデータとの間隔をpixel単位で指定します。

CELLPADDING属性を省略した場合、Webブラウザなどアプリケーション側で適当な値が与えられます。また、Lynxなどのテキストブラウザには枠が表現できないものがあり、その場合はCELLPADDING属性も表現できません。

CELLPADDING属性の指定

```
<TABLE BORDER="2" CELLPADDING="20">
<TR><TH>A</TH><TH>B</TH><TH>C</TH><TH>D</TH></TR>
<TR><TD>1</TD><TD>2</TD><TD>3</TD><TD>4</TD></TR>
<TR><TD>5</TD><TD>6</TD><TD>7</TD><TD>8</TD></TR>
</TABLE>
```



CELLPADDING属性の使用例

●BORDERCOLOR属性

表の枠色を指定します。色は16進数、あるいは色名で指定します。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●BORDERCOLORLIGHT属性

表の枠のハイライト色を指定します。色は16進数、あるいは色名で指定します。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●BORDERCOLORDARK属性

表の枠のシャドー色を指定します。色は16進数、あるいは色名で指定します。
Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

CAPTION <CAPTION>~</CAPTION>

表（table or figure CAPTION） ブロックレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト
※0個以上

親要素 ▶ TABLE

要素の概要

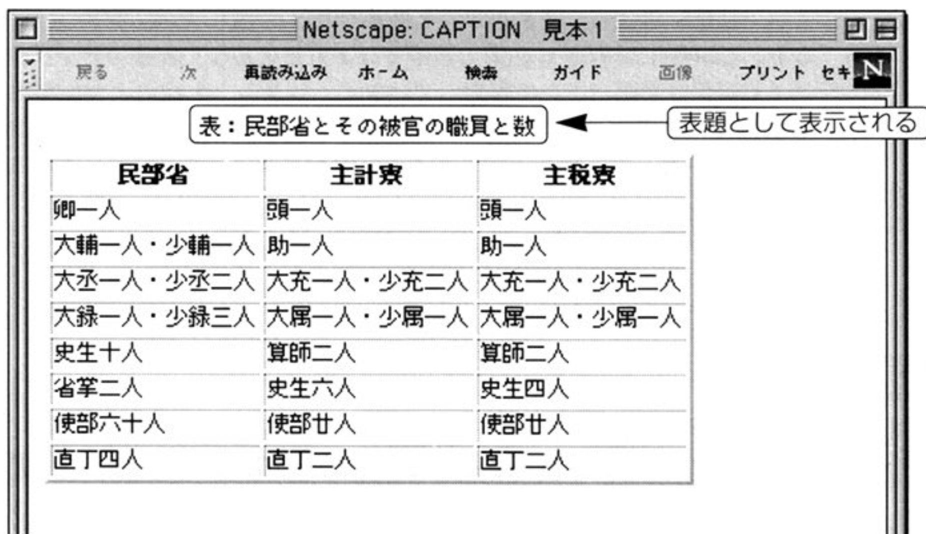
CAPTION要素は、表題のための要素です。

開始タグ<CAPTION>と終了タグ</CAPTION>によって、表題をマークアップします。内容としてはテキストレベル要素と一般的なテキストのみ置くことができます。

CAPTION要素は必須要素ではありませんが、使用する場合はTABLE要素の中の一番最初に1つだけ置かれます。他の要素の中にCAPTION要素を置くこと、あるいは1つのTABLE要素の中に複数のCAPTION要素を置くことはできません。

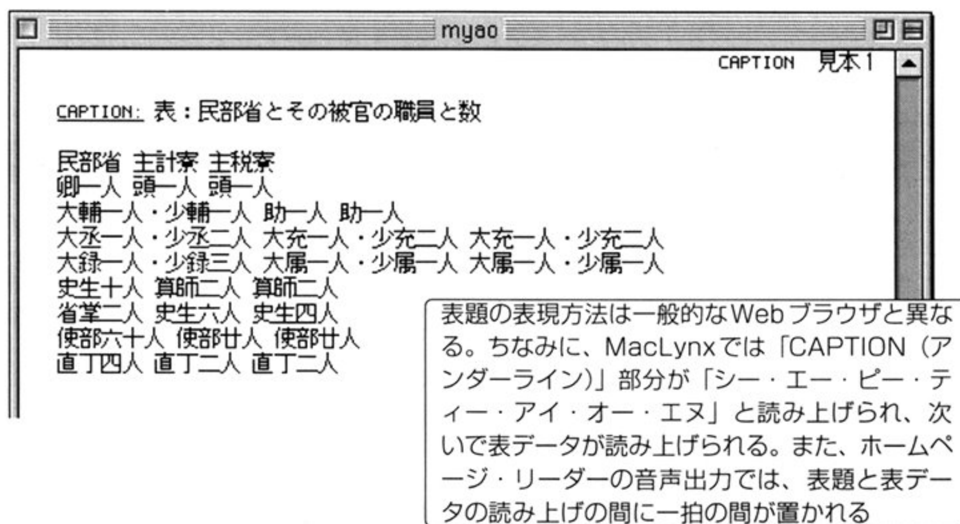
```
<TABLE BORDER="2">                                    CAPTION要素
<CAPTION>表：民部省とその被官の職員と数</CAPTION> ◀
<TR><TH>民部省</TH><TH>主計寮</TH><TH>主税寮</TH></TR>
<TR><TD>卿一人</TD><TD>頭一人</TD><TD>頭一人</TD></TR>
<TR><TD>大輔一人・少輔一人</TD><TD>助一人</TD><TD>助一人
</TD></TR>
<TR><TD>大丞一人・少丞二人</TD><TD>大充一人・少充二人</TD><TD>
大充一人・少充二人</TD></TR>
<TR><TD>大録一人・少録三人</TD><TD>大属一人・少属一人</TD><TD>
大属一人・少属一人</TD></TR>
<TR><TD>史生十人</TD><TD>算師二人</TD><TD>算師二人
</TD></TR>
<TR><TD>省掌二人</TD><TD>史生六人</TD><TD>史生四人
</TD></TR>
<TR><TD>使部六十人</TD><TD>使部廿人</TD><TD>使部廿人
</TD></TR>
<TR><TD>直丁四人</TD><TD>直丁二人</TD><TD>直丁二人
```

```
</TD></TR>
</TABLE>
```



CAPTION要素の使用例

表



Lynxでの表示例

注意 CAPTION要素が表の前（上）にあれば、Lynxなど表の形を完全に再現できないWebブラウザでも、以下に続くテキストが「表データ」であることが画面上でも確認できますし、音声出力機能ではなおさらです。CAPTION要素を必ずTR要素の前に置く理由、あるいはLynxでALIGN属性が表現されない（後述）理由は、その辺りにもあるのでしょう。
なお、CAPTION要素は必須の要素ではありませんし、本書のサンプルソースでも煩雑さを避けるため省略しています。しかし、上記のような理由から、本物のHTML文書ではできるだけ記述しましょう。また、その際に表題の先頭に「表」などの一言を入れておくと、表があることがよりはっきりするでしょう。

属性一覧

●ALIGN（初期値：top）

値	働き	H3	N4	I4
top	表の上部に表示	○	○	○
bottom	表の下部に表示	○	○	○
left	表の左側に揃える	−	−	!
center	表の中央に揃える	−	−	!
right	表の右側に揃える	−	−	!

属性は省略可

●VALIGN

値	働き	H3	N4	I4
top	表の上部に表示	−	−	!
bottom	表の下部に表示	−	−	!

属性は省略可

属性解説

●ALIGN属性

一般的な文書では、表題は表の上部に配置されますが、ALIGN属性を指定して、下部に指定することもできます。ただし、Lynxなどテキストブラウザでは、ALIGN属性は無視され、必ず表の上部に表示されます（おそらく表の認識性を高めるためでしょう）。

ALIGN属性の指定

```
<TABLE BORDER="2">
<CAPTION ALIGN="bottom">表：民部省とその被官の職員と数
</CAPTION>
<TR><TH>民部省</TH><TH>主計寮</TH><TH>主税寮</TH></TR>
<TR><TD>卿一人</TD><TD>頭一人</TD><TD>頭一人</TD></TR>
<TR><TD>大輔一人・少輔一人</TD><TD>助一人</TD><TD>助一人
</TD></TR>
(中略)
<TR><TD>直丁四人</TD><TD>直丁二人</TD><TD>直丁二人
</TD></TR>
</TABLE>
```

Netscape: CAPTION 見本2

戻る 次 再読み込み ホーム 検索 ガイド 画像 プリント セキ

民部省	主計寮	主税寮
卿一人	頭一人	頭一人
大輔一人・少輔一人	助一人	助一人
大丞一人・少丞二人	大充一人・少充二人	大充一人・少充二人
大録一人・少録三人	大属一人・少属一人	大属一人・少属一人
史生十人	算師二人	算師二人
省掌二人	史生六人	史生四人
使部六十人	使部廿人	使部廿人
直丁四人	直丁二人	直丁二人

表：民部省とその被官の職員と数

ALIGN属性により、CAPTION要素を下部に表示

ALIGN属性の使用例

なお、属性値のうち、left、center、rightはInternet Explorerの独自拡張です。これらはVALIGN属性と組み合わせて使用されます。

●VALIGN属性

ALIGN属性の属性値のうち、left、center、rightと組み合わせて、表題の位置を揃えるために使用されます。しかし、Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

TR

<TR>~</TR>

表の行要素（Table Row）　ブロックレベル要素

内容 ▶ TH　TD

親要素 ▶ TABLE

要素の概要

TR要素は表の行要素です。開始タグ<TR>と終了タグ</TR>によって表の行要素をマークアップします。終了タグは省略可能ですが、HTML文書が煩雑にならなければ残しておいた方がいいでしょう。

表の基本構成要素ですから、必ずTABLE要素の中に置かれます（他の要素の中にTR要素を置くことはできません）。内容としては表の見出しセルであるTH要素（P.112参照）とデータセルであるTD要素（P.122参照）のみを置くことができます。また、同じ行にTH要素とTD要素の両方を置くこともできます。

```
<TABLE BORDER="2">
<CAPTION>民部省とその被官の職員と数</CAPTION>
<TR><TH>民部省</TH><TH>主計寮</TH><TH>主税寮</TH></TR>
<TR><TD>卿一人</TD><TD>頭一人</TD><TD>頭一人</TD></TR>
<TR><TD>大輔一人・少輔一人</TD><TD>助一人</TD><TD>助一人
</TD></TR>
(中略)
<TR><TD>直丁四人</TD><TD>直丁二人</TD><TD>直丁二人
</TD></TR>
</TABLE>
```

属性一覧

●ALIGN

値	働き	H3	N4	I4
left	セルの左側に揃える	○	○	○
center	セルの中央に揃える	○	○	○
right	セルの右側に揃える	○	○	○

属性は省略可

●VALIGN

値	働き	H3	N4	I4
top	セルの上部に揃える	○	○	○
middle	セルの中央に揃える	○	○	○
bottom	セルの下部に揃える	○	○	○
baseline	テキストのベースラインに揃える	－	！	！

属性は省略可

属性解説

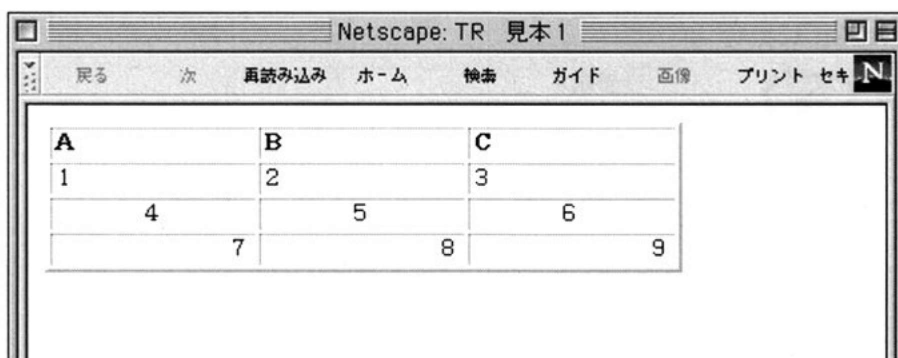
●ALIGN属性

セル内の見出しやデータの左右方向の位置揃えを指定します。表の幅を特定の幅に設定している場合や（TABLE要素のWIDTH要素）、セル内のデータの長さが異なる場合などに使用します。

属性値にはleft、center、rightの3種類があります。属性を省略した場合は、表の見出し（TH要素）はcenter、セル内のデータはleftが適用されます。

ALIGN属性の指定

```
<TABLE WIDTH="80%">
<TR ALIGN="left"><TH>A</TH><TH>B</TH><TH>C</TH></TR>
<TR ALIGN="left"><TD>1</TD><TD>2</TD><TD>3</TD></TR>
<TR ALIGN="center"><TD>4</TD><TD>5</TD><TD>6</TD></TR>
<TR ALIGN="right"><TD>7x</TD><TD>8</TD><TD>9</TD></TR>
</TABLE>
```



ALIGN属性の使用例

表

●VALIGN属性

セル内の見出しやデータの上下方向の位置揃えを指定します。セルの高さが中のデータより高い場合などに使用します。

属性値にはtop、middle、bottom、baselineの4種類があります。属性を省略した場合は、通常、middleが適用されます。また、Lynxなどテキストブラウザでは VALIGN属性は表現できません。

なお、属性値のうちbaselineはNetscape NavigatorとInternet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

VALIGN属性の指定

```
<TABLE BORDER=2 WIDTH="80%">
<TR VALIGN="top"><TD><IMG SRC="tel1.gif"></TD><TD>上部揃え</TD></TR>
<TR VALIGN="middle"><TD><IMG SRC="tel1.gif"</TD><TD>中央揃え</TD></TR>
<TR VALIGN="bottom"><TD><IMG SRC="tel1.gif"></TD><TD>下部揃え</TD></TR>
<TR VALIGN="baseline"><TD><IMG SRC="tel1.gif"></TD><TD>ベースライン揃え</TD></TR>
</TABLE>
```

※見やすくするためにIMG要素のWIDTH・HEIGT・ALT属性を省略



VALIGN属性の使用例

TH

<TH>~</TH>

表の見出し要素 (Table Header cell) **ブロックレベル要素**

内容 H1 H2 H3 H4 H5 H6
ADDRESS
P UL OL DL (DIR) (MENU) PRE DIV CENTER
BLOCKQUOTE FORM ISINDEX HR TABLE
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 TR

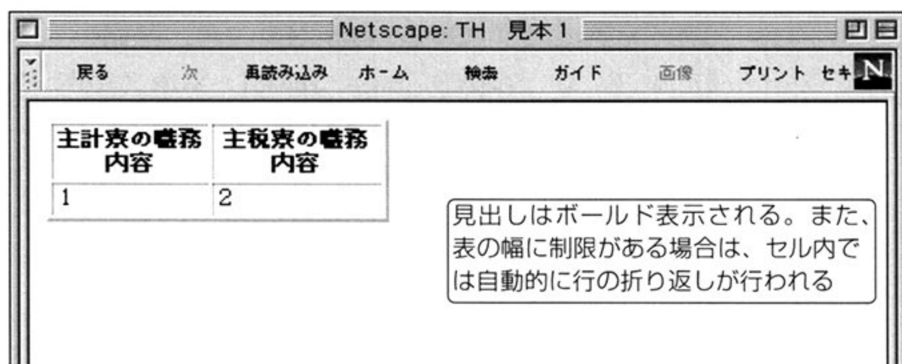
要素の概要

TH要素は表の見出し要素です。開始タグ<TH>と終了タグ</TH>によって、表の見出し要素をマークアップします。終了タグは省略可能ですが、HTML文書が煩雑にならなければ残しておいた方がいいでしょう。

Webブラウザでは、見出しのテキストはボールド表示され、データと区別されます。また、セルの幅は同じTABLE要素内の最大のセルに自動的に調整されます。あるいは、表の幅が特定の値に設定されている場合は(TABLE要素のWIDTH属性)、自動的にセル内でテキストが折り返しされます。一方、Lynxなどテキストブラウザにはこれらの表示・調整機能がないものもあり、その場合は見かけ上表の形は崩れて表示されます。とはいっても、セルの区切りは空白文字などに置換されて表現されるため、データを保存して別のアプリケーションで加工表示することはできます。

表の行を構成する要素ですから、必ずTR要素の中に置かなければなりません(他の要素の中にTH要素を置くことはできません)。内容としてはテキストレベル要素や一般的なテキストだけでなく、ブロックレベル要素も置くことができます。あまり使用する機会はないと思いますが、表の入れ子も可能です。

```
<TABLE BORDER="2" WIDTH="45%">
<TR><TH>主計寮の職務内容</TH><TH>主税寮の職務内容</TH></TR>
<TR><TD>1</TD><TD>2</TD</TR>
</TABLE>
```



TH要素の表示例

属性一覧

●NOWRAP

値	働き	H3	N4	I4
nowrap	セル内での折り返し禁止	○	○	○

属性は省略可

●ROWSPAN（初期値：1）

値	働き	H3	N4	I4
number	セルの高さを行数で指定	○	○	○

属性は省略可。0より大きい数値

●CPLSPAN（初期値：1）

値	働き	H3	N4	I4
number	セルの幅を列数で指定	○	○	○

属性は省略可。0より大きい数値

●ALIGN

値	働き	H3	N4	I4
left	セルの左側に揃える	○	○	○
center	セルの中央に揃える	○	○	○
right	セルの右側に揃える	○	○	○

属性は省略可

●VALIGN

値	働き	H3	N4	I4
top	セルの上部に揃える	○	○	○
middle	セルの中央に揃える	○	○	○
bottom	セルの下部に揃える	○	○	○
baseline	テキストのベースラインに揃える	−	!	!

属性は省略可

●WIDTH

値	働き	H3	N4	I4
pixels	セルの幅	○	○	○

属性は省略可

●HEIGHT

値	働き	H3	N4	I4
pixels	表の高さ	○	○	○

属性は省略可

●BGCOLOR

値	働き	H3	N4	I4
color	セルの背景色を指定（16進数、色名）	−	!	!

属性は省略可



color=色指定の16進数（#rrggbb）または色名　*pixels*=pixel数
number=0より大きい数値

属性解説

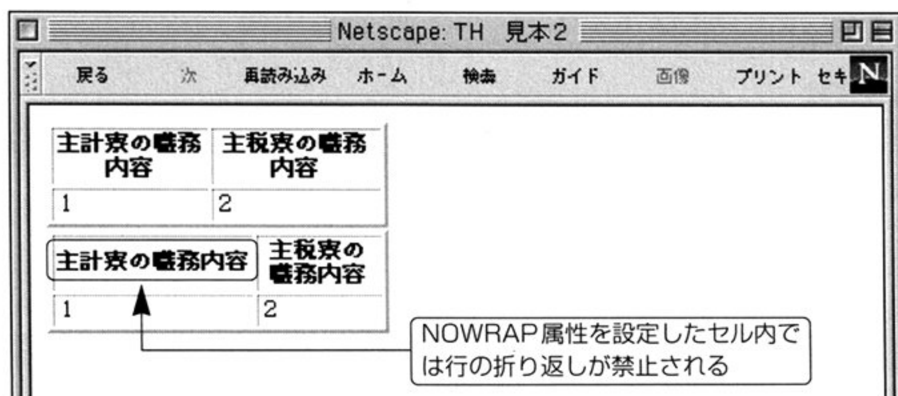
●NOWRAP属性

Webブラウザでは、表の表示幅をウインドウの幅に合わせて自動調整します。また、それに合わせてセル内のデータも自動的に折り返されます。しかし、NOWRAP属性を指定することによって、セル内の自動折り返しを禁止できます。ただし、テキストブラウザではNOWRAP属性を表現できないものもあります。

表

NOWRAP属性を指定する場合は、属性名は省略し属性値「nowrap」のみを記述します(P.8参照)。

```
<TABLE BORDER="2" WIDTH="45%">
<TR><TH>主計寮の職務内容</TH><TH>主税寮の職務内容</TH></TR>
<TR><TD>1</TD><TD>2</TD</TR>
</TABLE>
<TABLE BORDER="2" WIDTH="45%">
<TR><TH nowrap>主計寮の職務内容</TH><TH>主税寮の職務内容
</TH></TR>
<TR><TD>1</TD><TD>2</TD</TR>
</TABLE>
```
NOWRAP属性の指定



NOWRAP属性の使用例

●ROWSPAN属性

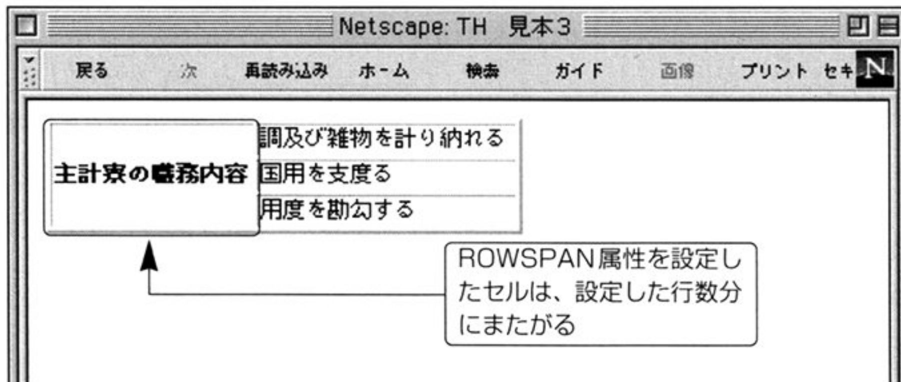
表内の一部で複数行にまたがるセルを作成する場合に使用する属性です。

属性値には行数(0より大きい整数)を指定します。また、表全体の行数を超える数値を指定することはできません。属性を省略した場合は、自動的に値1が与えられます。

なお、Lynxなどテキストブラウザには ROWSPAN属性を表現できないものもあります。また、音声出力では表のイメージが得られにくくなるため、使用に当たっては注意が必要です。

```
<TABLE BORDER="2" WIDTH="60%">
    <TR><TH ROWSPAN="3">主計寮の職務内容</TH><TD>調及び雑物
を計り納れる</TD></TR>
    <TR><TD>国用を支度る</TD></TR>
    <TR><TD>用度を勘勾する</TD></TR>
</TABLE>
```

ROWSPAN属性の指定



ROWSPAN属性の使用例

●COLSPAN属性

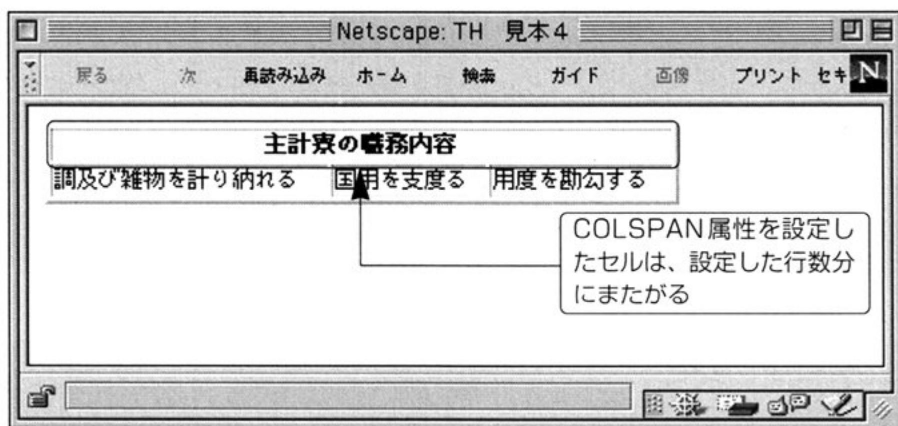
表内の一部で複数列にまたがるセルを作成する場合に使用する属性です。

属性値には列数（0より大きい数値）を指定します。また、表全体の列数を超える数値を指定することはできません。属性を省略した場合は、自動的に値1が設定されます。

なお、Lynxなどテキストブラウザには COLSPAN属性を表現できないものもあります。また、音声出力では表のイメージが得られにくくなるため、使用に当たっては注意が必要です。

```
<TABLE BORDER="2" WIDTH="60%">
<TR><TH COLSPAN="3">主計寮の職務内容</TH></TR>
<TR><TD>調及び雑物を計り納れる</TD><TD>国用を支度る</TD><TD>
用度を勘勾する</TD></TR>
</TABLE>
```

COLSPAN属性の指定



COLSPAN属性の使用例

●ALIGN属性

セル内の見出しの左右方向の位置揃えを指定します。表の幅を特定の幅に設定している場合や(TABLE要素のWIDTH要素)、セル内のデータの長さが異なる場合などに使用します。

属性値にはleft、center、rightの3種類があります。属性を省略した場合は、centerが適用されます。また、Lynxなどテキストブラウザでは WIDTH属性を表現できないものもあり、その場合はALIGN属性も表現できません。

```
<TABLE BORDER="2" WIDTH="80%">
<TR><TH ALIGN="left"> A </TH><TH ALIGN="center"> B
</TH><TH ALIGN="right"> C </TH></TR>
<TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD></TR>
<TR><TD> 4 </TD><TD> 5 </TD><TD> 6 </TD></TR>
<TR><TD> 7 </TD><TD> 8 </TD><TD> 9 </TD></TR>
</TABLE>
```

ALIGN属性の指定

Netscape: TH 見本5

戻る 次 再読み込み ホーム 検索 ガイド 画像 プリント セキ

A	B	C
1	2	3
4	5	6
7	8	9

ALIGN属性の使用例

●VALIGN属性

セル内の見出しの上下方向の位置揃えを指定します。セルの高さが中のデータより高い、たとえば、同じ行にIMG要素が置かれていたり、HEIGHT属性が設定されている場合などに使用します（サンプルソースと表示結果はHEIGHT属性の項を参照）。

属性値にはtop、middle、bottom、baselineの4種類があります。属性を省略した場合は、通常、middleが適用されます。また、Lynxなどテキストブラウザでは VALIGN属性は表現できません。

なお、属性値のうちbaselineはNetscape NavigatorとInternet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

表

●WIDTH属性

セルの幅をpixel単位で指定します。

また、Lynxなどテキストブラウザには WIDTH属性を表現できないものもあります。

```
<TABLE BORDER="2">
<TR><TH WIDTH="40"> A </TH><TH WIDTH="80"> B </TH><TH
WIDTH="20"> C </TH></TR>
<TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD></TR>
</TABLE>
```

WIDTH属性の指定

Netscape: TH 見本6

A	B	C
1	2	3

40 pixel　80 pixel　20 pixel

WIDTH属性の使用例

●HEIGHT属性

セルの高さをpixel単位で指定します。また、セル内の見出しの上下方向の位置揃えをしたい場合は、VALIGN属性を併用します。

なお、理屈上同じ行で違う高さを指定することはできません。また、Lynxなどテキストブラウザには、HEIGHT属性を表現できないものもあります。

```
<TABLE BORDER="2" WIDTH="80%">
<TR><TH HEIGHT="40" VALIGN="top">A</TH><TH
HEIGHT="40" VALIGN="middle">B</TH><TH HEIGHT="40"
VALIGN="bottom">C</TH></TR>
<TR><TD>1</TD><TD>2</TD><TD>3</TD></TR>
</TABLE>
```

Netscape: TH 見本7

A	B	C
1	2	3

高さ（HEIGHT="40"）
上部揃え（VALIGN="top"）
中央揃え（VALIGN="middle"）
下部揃え（VALIGN="bottom"）

HEIGHT要素の使用例

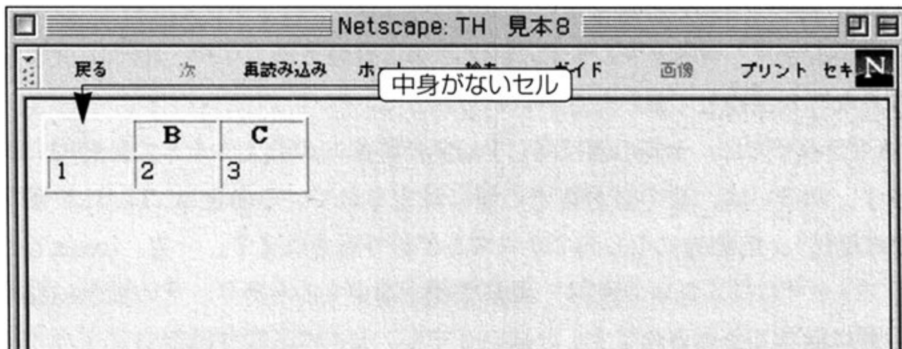
●BGCOLOR属性

セルごとの背景色を指定します。色は16進数、あるいは色名で指定します。

Netscape NavigatorとInternet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

注意 表の内容によっては、見出しの中身がないセルを含む場合もあります。その場合、一般的なWebブラウザではセルの枠が表示されません。なお、音声出力では表のイメージが得られにくくなるため、使用にあたっては注意が必要です。

```
<TABLE BORDER="2">
<TR><TH WIDTH="40"></TH><TH WIDTH="40">B</TH><TH
WIDTH="40">C</TH></TR>
<TR><TD>1</TD><TD>2</TD><TD>3</TD></TR>
</TABLE>
```



データがない場合の表示例

NN 3 IE 3 NN 4 IE 4

TD

<TD>～</TD>

表の見出し要素（Table Data cell）　ブロックレベル要素

内容 H1 H2 H3 H4 H5 H6
ADDRESS
P UL OL DL (DIR) (MENU) PRE DIV CENTER
BLOCKQUOTE FORM ISINDEX HR TABLE
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 TR

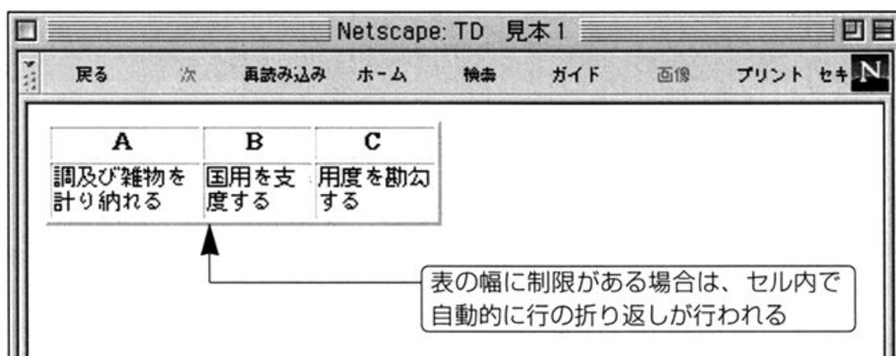
要素の概要

TD要素は表のデータ要素です。開始タグ<TD>と終了タグ</TD>によって表のデータ要素をマークアップします。終了タグは省略可能ですが、HTML文書が煩雑にならなければ残しておいた方がいいでしょう。

Webブラウザでは、セルの幅は同じTABLE要素内の最大のセルに自動的に調整されます。あるいは、表の幅が特定の値に設定されている場合は（TABLE要素のWIDTH属性）、自動的にセル内でテキストが折り返されます。一方、Lynxなどテキストブラウザにはこれらの表示・調整機能がないものもあり、その場合は見かけ上表の形は崩れて表示されます。とはいっても、セルの区切りは空白文字などに置換されて表現されるため、データを保存して別のアプリケーションで加工・表現することはできます。

表の行を構成する要素ですから、必ずTR要素の中に置かれます（他の要素の中にTD要素を置くことはできません）。内容としてはテキストレベル要素や一般的なテキストだけでなく、ブロックレベル要素も置くことができます（あまり使用する機会はないと思いますが、表の入れ子も可能です）。

```
<TABLE BORDER="2" WIDTH="50%">
<TR><TH>A</TH><TH>B</TH><TH>C</TH></TR>
<TR><TD>調及び雑物を計り納れる</TD><TD>国用を支度する</TD>
<TD>用度を勘勾する</TD></TR>
</TABLE>
```



TD要素の使用例

属性一覧

●NOWRAP

値	働き	H3	N4	I4
nowrap	セル内での折り返し禁止	○	○	○

属性は省略可

●ROWSPAN（初期値：１）

値	働き	H3	N4	I4
number	セルの高さを行数で指定	○	○	○

属性は省略可。0より大きい数値

●COLSPAN（初期値：１）

値	働き	H3	N4	I4
number	セルの幅を列数で指定	○	○	○

属性は省略可。0より大きい数値

●ALIGN

値	働き	H3	N4	I4
left	セルの左側に揃える	○	○	○
center	セルの中央に揃える	○	○	○
right	セルの右側に揃える	○	○	○

属性は省略可

●VALIGN

値	働き	H3	N4	I4
top	セルの上部に揃える	○	○	○
middle	セルの中央に揃える	○	○	○
bottom	セルの下部に揃える	○	○	○
baseline	テキストのベースラインに揃える	−	!	!

属性は省略可

●WIDTH

値	働き	H3	N4	I4
pixels	セルの幅	○	○	○

属性は省略可

●HEIGHT

値	働き	H3	N4	I4
pixels	表の高さ	○	○	○

属性は省略可

●BGCOLOR

値	働き	H3	N4	I4
color	セルの背景色を指定	−	!	!

属性は省略可

変数　*color*=色指定の16進数（#rrggbb）または色名
pixels=pixel数　　*number*=0より数値

属性解説

●NOWRAP属性

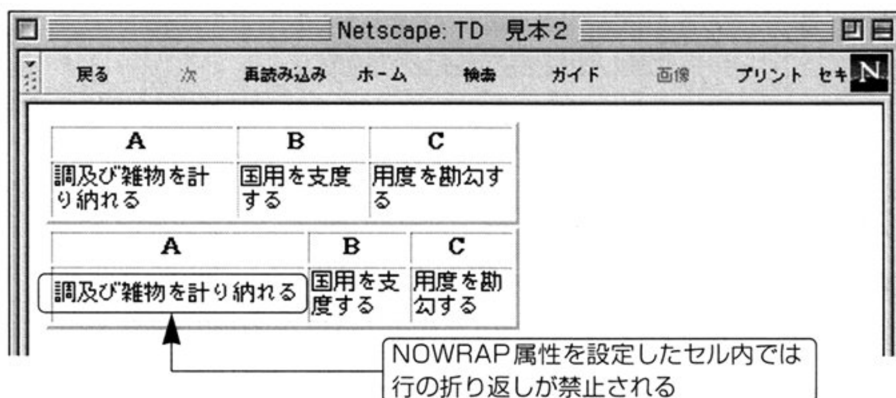
Webブラウザでは、表の表示幅をウインドウの幅に合わせて自動調整します。また、それに合わせてセル内のデータも自動的に折り返されます。しかし、NOWRAP属性を指定することによって、セル内の自動折り返しを禁止できます。また、Lynx

などテキストブラウザにはNOWRAP属性を表現できないものもあります。

NOWRAP属性を指定する場合は、属性名は省略し属性値「nowrap」のみを記述します(P.8参照)。

```
<TABLE BORDER="2" WIDTH="60%">
<TR><TH> A </TH><TH> B </TH><TH> C </TH></TR>
<TR><TD>調及び雑物を計り納れる</TD><TD>国用を支度する
</TD><TD>用度を勘勾する</TD></TR>
</TABLE>
<TABLE BORDER="2" WIDTH="60%">
<TR><TH> A </TH><TH> B </TH><TH> C </TH></TR>
<TR><TD nowrap>調及び雑物を計り納れる</TD><TD>国用を支度する
</TD><TD>用度を勘勾する</TD></TR>
</TABLE>
```

NOWRAP属性を指定



NOWRAP属性の使用例

●ROWSPAN属性

表内の一部で複数行にまたがるセルを作成する場合に使用する属性です。

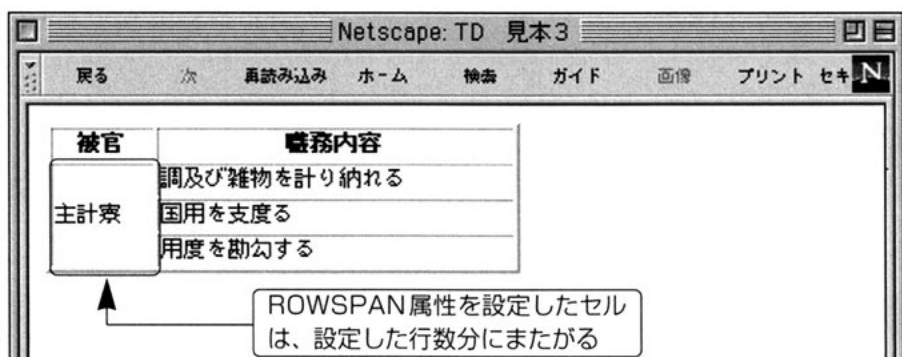
属性値には行数(0より大きい数値)を指定します。また、表全体の行数を超える数値を指定することはできません。属性を省略した場合は、自動的に値1が与えられます。

なお、Lynxなどテキストブラウザには ROWSPAN属性を表現できないものもありますし、音声出力では表のイメージが得られにくくなるため、使用に当たっては注意が必要です。

表

```
<TABLE BORDER="2" WIDTH="60%">
    <TR><TH>被官  </TH><TH>職務内容  </TH></TR>
    <TR><TD ROWSPAN="3">主計寮</TD><TD>調及び雑物を計り納れ
る</TD></TR>
<TR><TD>国用を支度る</TD></TR>
<TR><TD>用度を勘勾する</TD></TR>
</TABLE>
```

ROWSPANの属性を指定



ROWSPAN属性の使用例

●COLSPAN属性

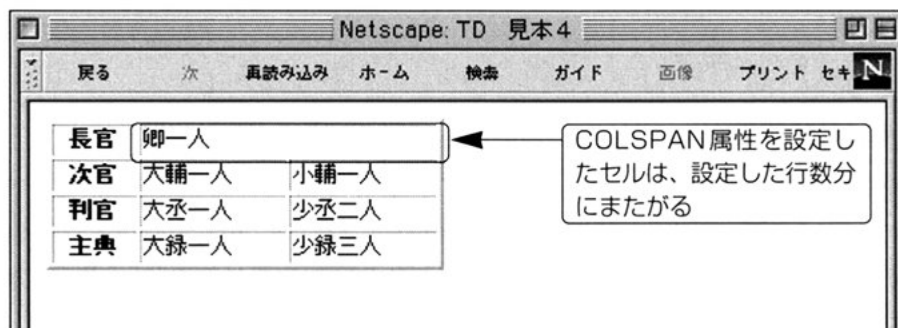
表内の一部で、複数列にまたがるセルを作成する場合に使用する属性です。

属性値には列数（0より大きい数値）を指定します。また、表全体の列数を超える数値を指定することはできません。属性を省略した場合は、自動的に値1が与えられます。

なお、Lynxなどテキストブラウザには COLSPAN属性を表現できないものもあります。また、音声出力では表のイメージが得られにくくなるため、使用に当たっては注意が必要です。

```
<TABLE BORDER="2" WIDTH="50%">
<TR><TH>長官</TH><TD COLSPAN="2">卿一人</TD></TR>
<TR><TH>次官</TH><TD>大輔一人</TD><TD>小輔一人</TD></TR>
<TR><TH>判官</TH><TD>大丞一人</TD><TD>少丞二人</TD></TR>
<TR><TH>主典</TH><TD>大録一人</TD><TD>少録三人</TD></TR>
</TABLE>
```

CLOSPAN属性の指定



COLSPAN属性の使用例

●ALIGN属性

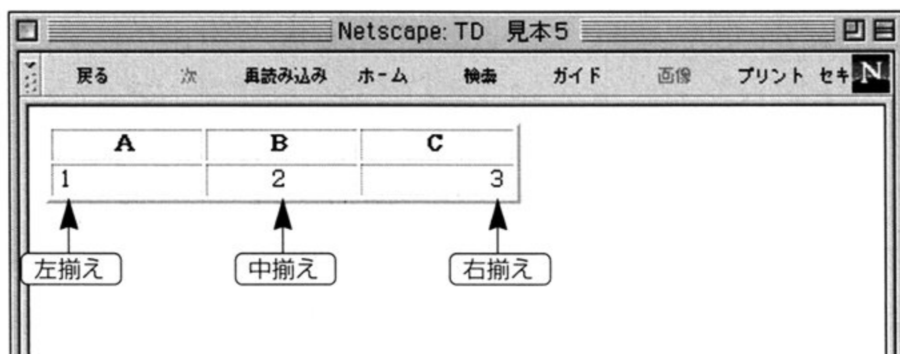
セル内のデータの左右方向の位置揃えを指定します。表の幅を特定の幅に設定している場合や(TABLE要素のWIDTH要素)、セル内のデータの長さが異なる場合などに使用します。

属性値にはleft、center、rightの3種類があります。属性を省略した場合は、Webブラウザではcenterが適用されます。また、Lynxなどテキストブラウザでは WIDTH属性を表現できないものもあり、その場合はALIGN属性も表現できません。

```
<TABLE BORDER="2" WIDTH="60%">
<TR><TH>A          </TH><TH>B          </TH><TH>C
</TH></TR>
<TR><TD ALIGN="left">1</TD><TD ALIGN="center">2
</TD><TD ALIGN="right">3</TD></TR>
</TABLE>
```

ALIGN属性の設定

ALIGN属性の設定



ALIGN属性の使用例

●VALIGN属性

セル内の見出しの上下方向の位置揃えを指定します。セルの高さが中のデータより高い、たとえば、同じ行にIMG要素が置かれていたり、HEIGHT属性が設定されている場合などに使用します(サンプルソースと表示結果はHEIGHT属性の項を参照)。

属性値にはtop、middle、bottom、baselineの4種類があります。属性を省略した場合は、Webブラウザでは、middleが適用されます。また、LynxなどテキストブラウザではVALIGN属性は表現できません。

なお、属性値のうちbaselineはNetscape NavigatorとInternet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●WIDTH属性

セルの幅をpixel単位で指定します。

また、Lynxなどテキストブラウザには WIDTH属性を表現できないものもあります。

```
<TABLE BORDER="2">
<TR><TH> A </TH><TH> B </TH><TH> C </TH></TR>
<TR><TD WIDTH="40"> 1 </TD><TD WIDTH="80"> 2 </TD><TD
WIDTH="20"> 3 </TD></TR>
</TABLE>
```

WIDTH属性の設定

WIDTH属性の使用例

●HEIGHT属性

セルの高さをpixel単位で指定します。また、セル内の見出しの上下方向の位置揃えをしたい場合は、VALIGN属性を併用します。

なお、理屈上同じ行で違う高さを指定することはできません。また、Lynxなどテキストブラウザには HEIGHT属性を表現できないものもあります。

```
<TABLE BORDER="2" WIDTH="80%">
<TR><TH> A </TH><TH> B </TH><TH> C </TH></TR>
<TR><TD HEIGHT="40" VALIGN="top"> 1 </TD><TD HEIGHT="40"
VALIGN="middle"> 2 </TD><TD HEIGHT="40" VALIGN="bot
tom"> 3 </TD></TR>
</TABLE>
```

HEIGHT属性の設定

HEIGHT属性の使用例

表

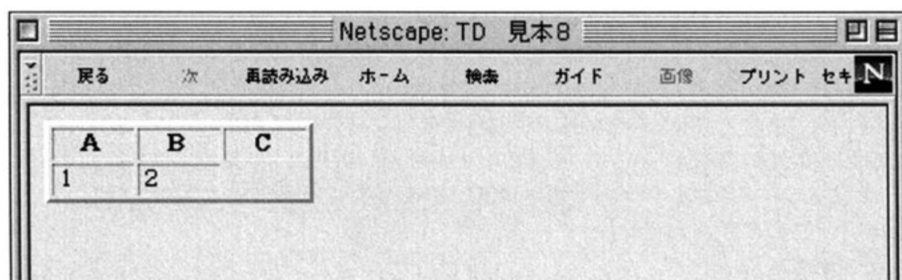
●BGCOLOR属性

セルごとの背景色を指定します。色は16進数、あるいは色名で指定します。

Netscape NavigatorとInternet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

注意 表の内容によっては、データがないセルを含む場合もあります。その場合、Webブラウザではセルの枠が表示されません。なお、音声出力では表のイメージが得られにくくなるため、使用にあたっては注意が必要です。

```
<TABLE BORDER="2">
<TR><TH>A</TH><TH>B</TH><TH>C</TH></TR>
<TR><TD WIDTH="40">1</TD><TD WIDTH="40">2</TD><TD
WIDTH="40"></TD></TR>
</TABLE>
```



データがない場合の表示例

NN 3 IE 3 NN 4 IE 4

FONT

<FONT>~</FONT>

フォント (FONT element)　　テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
IMG APPLET FONT BASEFONT BR MAP INPUT
SELECT TEXTAREA SCRIPT
一般的なテキスト

親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P LI DT DD DIV CENTER
BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
APPLET FONT

要素の概要

文章中のテキストのフォントサイズや色を部分的に指定する要素です。

開始タグ<FONT>と終了タグ</FONT>でマークアップされたテキストは、Webブラウザで表示する場合、指定されたサイズと色が使用されます。

属性一覧

●SIZE

値	働き	H3	N4	I4
number	サイズを絶対値で指定（1～7）	○	○	○
± *number*	サイズを相対値で指定（1～6）	○	○	○

属性は省略可

●COLOR

値	働き	H3	N4	I4
color	テキスト色を指定（16進数、色名）	○	○	○

属性は省略可

●FACE

値	働き	H3	N4	I4
fontname	フォント名を指定	−	!	!

属性は省略可



number=数値　*color*=色名　*fontname*=フォント名

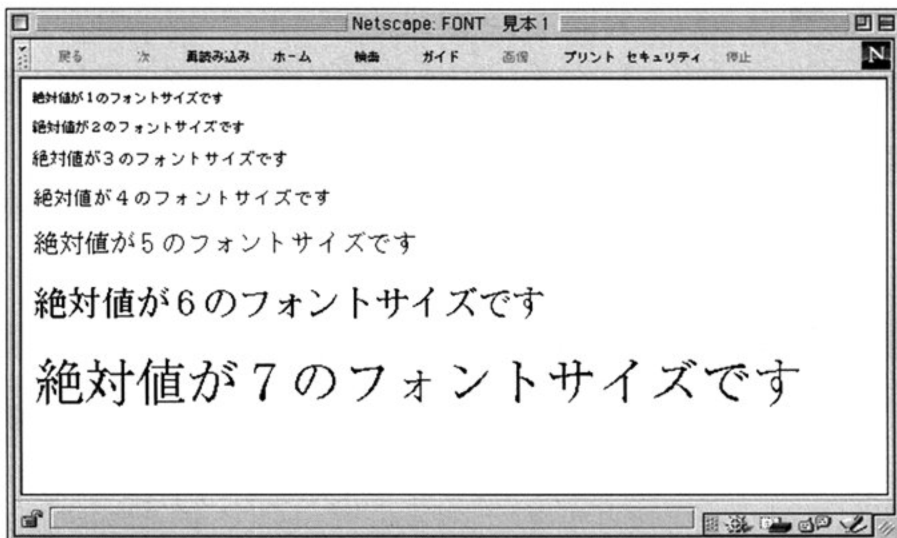
属性解説

●SIZE属性

フォントサイズの指定には、絶対値による指定と相対値による指定の2種類の方法があります。

絶対値によって指定する場合は、1～7の整数を用います。ただし、「絶対値」といっても一般のワープロやDTPのように正確に「○ポイント」と指定できるわけではありません。実際に表示されるフォントサイズは、Webブラウザ側で設定されたサイズが基準になります。そのため、極端なサイズを指定すると、見る人によっては小さくて読めなくなったり、あるいは大きすぎて読みづらくなるので、ユーザが特定できる場合を除いて積極的な使用はおすすめできません。

```
<P><FONT SIZE="1">絶対値が1のフォントサイズです</FONT></P>
<P><FONT SIZE="2">絶対値が2のフォントサイズです</FONT></P>
<P><FONT SIZE="3">絶対値が3のフォントサイズです</FONT></P>
<P><FONT SIZE="4">絶対値が4のフォントサイズです</FONT></P>
<P><FONT SIZE="5">絶対値が5のフォントサイズです</FONT></P>
<P><FONT SIZE="6">絶対値が6のフォントサイズです</FONT></P>
<P><FONT SIZE="7">絶対値が7のフォントサイズです</FONT></P>
```



SIZE属性の表示例

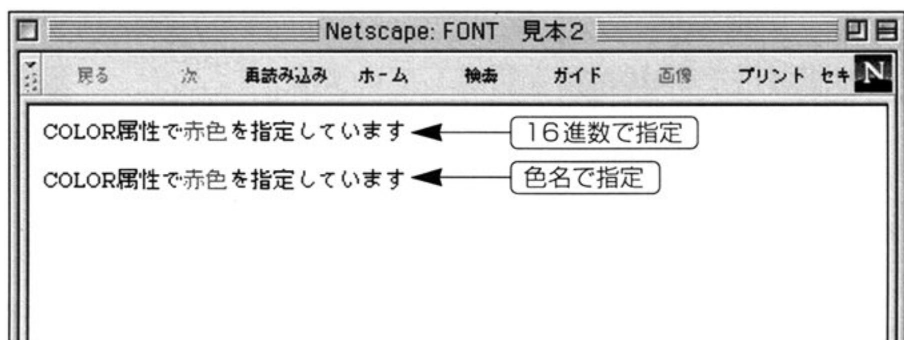
ただし、Lynxなどテキストブラウザでは、画面上ではSIZE属性を表現できません。

なお、相対値による指定については、BASEFONT要素（P.135）を参照してください。

●COLOR属性

Webブラウザで HTML文書を表示する場合、テキスト色を部分的に変更したい場合に使用します。色は16進数、あるいは色名で指定します（P.24参照）。

```
<P>COLOR属性で<FONT COLOR="##ff0000">赤色</FONT>を指定し
ています</P>
<P>COLOR属性で<FONT COLOR="red">赤色</FONT>を指定していま
す</P>
```



COLOR属性の表示例

なお、テキストに色を指定する場合、背景色との組み合わせによっては非常に見えにくくなる場合もあります。背景色はHTML文書側で指定がない場合、Webブラウザに設定してある色が使用されます。

もし必要があってCOLOR属性を使用する場合は、BODY要素のBGCOLOR属性（P.25参照）も含めて設定を行った方がいいかもしれません。ただし、人の視覚には差があります。ユーザによっては、自分に見やすいテキスト色と背景色をWebブラウザで設定している場合もあります。それをCOLOR属性やBGCOLOR属性で強制的に切り替えられると、かえって見えづらくなってしまう可能性もありますので、ユーザが特定できる場合を除いて積極的な使用はおすすめできません。

●FACE属性

Webブラウザで HTML 文書を表示する場合、表示フォントの種類を具体的なフォント名で指定する属性です。しかし、Netscape Navigator とInternet Explorerのみが解釈できる独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

```
<P>FACE属性で<FONT FACE="chicago">Chicago</FONT>を指定
しています</P>
<P>FACE属性で<FONT FACE="symbol">Symbol</FONT>を指定して
います</P>
<P>FACE属性で<FONT FACE="新ゴB">新ゴB</FONT>を指定しています
</P>
```



Netscape NavigatorでのFACE属性の使用例

また、ユーザが特定できる場合でも、作者が希望するフォントを全てのユーザがシステムにインストールしているとは限りません。したがって、この属性を使用する場合は、複数のフォント名を候補として挙げておくことができます。候補は「,」で区切って並記します。また、その候補に該当するフォント全てが存在しない場合は属性が無視されて、Webブラウザに設定されている標準フォントを使用して表示されます。

BASEFONT

<BASEFONT>（終了タグなし）

基準フォントサイズ（BASEFONT element）　テキストレベル要素

内容 ▶ 空

親要素 ▶ BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　LI　DT　DD　DIV　CENTER　BLOCKQUOTE
FORM　CAPTION　TH　TD
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
APPLET　FONT

要素の概要

文章中のテキストの基準となるフォントサイズを指定する要素で、通常はFONT要素のSIZE属性（相対値指定）と組み合わせて使用します。

BASEFONT要素は開始タグ<BASEFONT>のみでマークアップされる、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

属性一覧

●SIZE（初期値：3）

値	働き	H3	N4	I4
number	サイズを絶対値で指定（1～7）	○	○	○

属性は省略可


number＝数値

属性解説

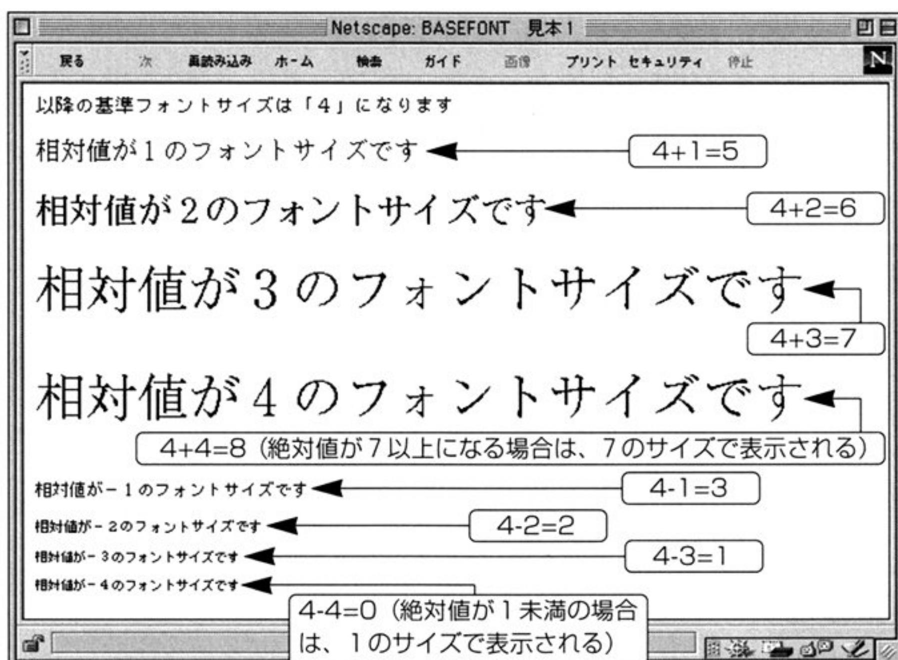
●SIZE属性

基準となるフォントサイズを指定します。値には1～7の整数が使用できます。

BASEFONT要素で基準フォントサイズを指定すると、それ以降のテキストにFONT要素のSIZE属性で相対値指定（値の前に「+」や「-」を付加）を行った場合、このフォントサイズを基にした相対値で表示されます。ただし、FONT要素のSIZE属性の絶対値指定（1～7）に換算して、それを超えるサイズは表示できません。

なお、実際に表示されるフォントサイズは、Webブラウザ側で設定されたサイズがさらに上位の基準になります。そのため、BASEFONT要素に極端なサイズを指定すると、見る人によっては小さくて読めなくなったり、あるいは大きすぎて読みづらくなるので、ユーザが特定できる場合を除いて積極的な使用はおすすめできません。

```
<P><BASEFONT SIZE="4">以降の基準フォントサイズは「4」になります
</P>
<P><FONT SIZE="+1">相対値が1のフォントサイズです</FONT></P>
<P><FONT SIZE="+2">相対値が2のフォントサイズです</FONT></P>
<P><FONT SIZE="+3">相対値が3のフォントサイズです</FONT></P>
<P><FONT SIZE="+4">相対値が4のフォントサイズです</FONT></P>
<P><FONT SIZE="-1">相対値が-1のフォントサイズです</FONT></P>
<P><FONT SIZE="-2">相対値が-2のフォントサイズです</FONT></P>
<P><FONT SIZE="-3">相対値が-3のフォントサイズです</FONT></P>
<P><FONT SIZE="-4">相対値が-4のフォントサイズです</FONT></P>
```



SIZE属性の設定例

TT

<TT>~</TT>

等幅フォント（TeleType or monospaced text style） テキストレベル要素

内容 TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

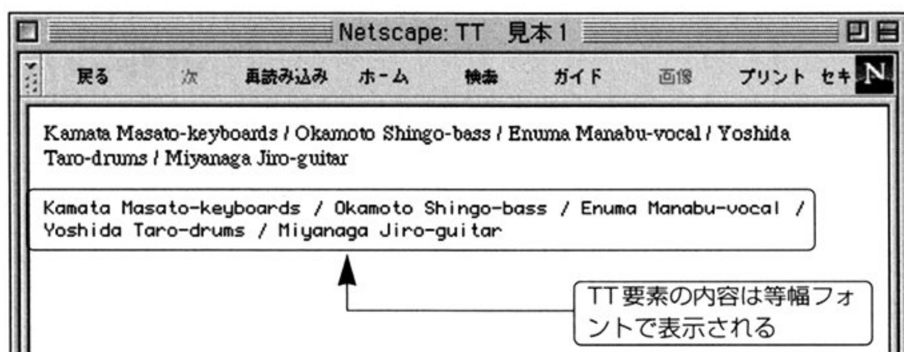
親要素 BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER
BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

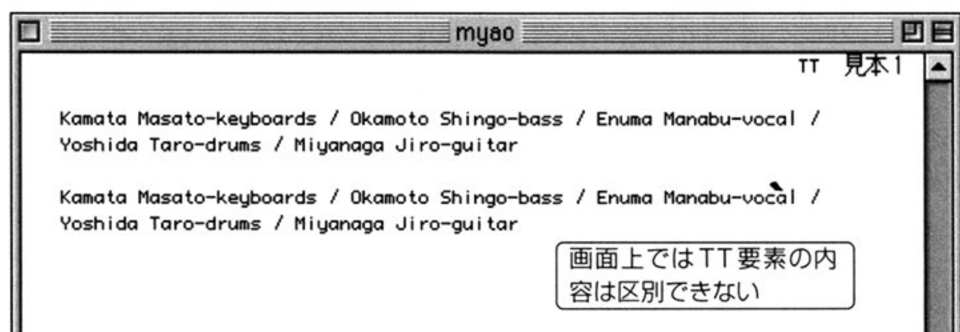
フォントスタイルを大まかに指定する要素です。

開始タグ<TT>と終了タグ</TT>でマークアップされたテキストは、Webブラウザで表示する場合、タイプライターでの出力をイメージするために、等幅フォント(として設定されているもの)が使用されます。

```
<P>Kamata Masato-keyboards / Okamoto Shingo-bass /
Enuma Manabu-vocal / Yoshida Taro-drums / Miyanaga
Jiro-guitar</P>
<P><TT>Kamata Masato-keyboards / Okamoto Shingo-bass
/ Enuma Manabu-vocal / Yoshida Taro-drums / Miyanaga
Jiro-guitar</TT></P>
```



TT要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

I

<I>～</I>

イタリック（Italic text style） テキストレベル要素

内容 TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET IMG FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

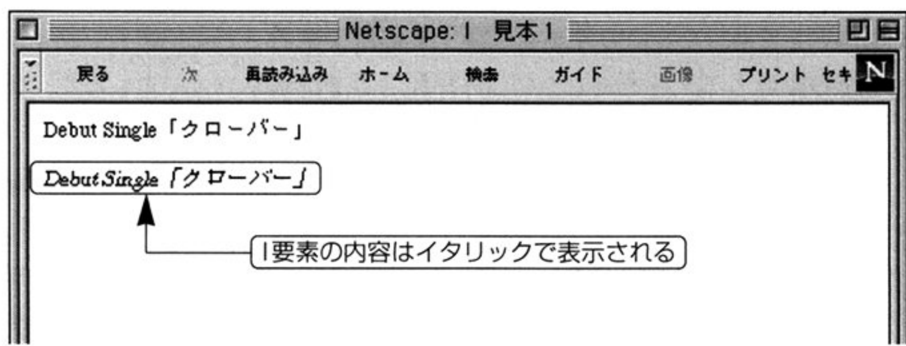
親要素 BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER
BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

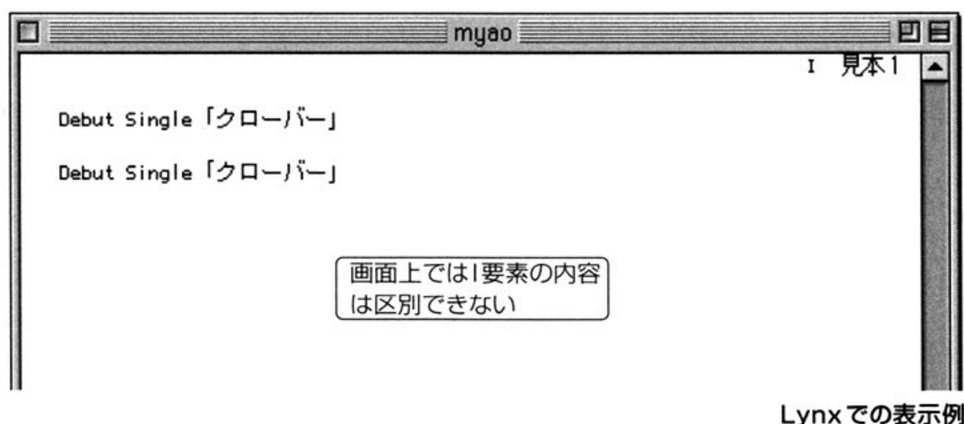
フォントスタイルを大まかに指定する要素です。

開始タグ<I>と終了タグ</I>でマークアップされたテキストは、イタリック表示されます。

```
<P>Debut Single「クローバー」</P>
<P><I>Debut Single「クローバー」</I></P>
```



I要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

Column　フォト関連の要素について

何度もくり返しになりますが、HTMLの本来の目的はテキストデータを使用することによって文書の互換性を維持しつつ、かつ文書の構造をマークアップすることによって、その情報を誰もがが間違いなく共有することにあります。

しかし、インターネット利用者の劇的な増加に伴って「見栄えのいいWebページを作りたい・見たい」という要求が多くなってきました。そこで、HTML 3.2ではいくつかの見栄えに関する要素が定義されています。フォント関連の要素もごく基本的なものはHTML 2.0で定義されていましたが、多くはHTML 3.2で追加されたものです。

HTML 3.2のフォント関連の要素を使用すれば、少なくともテキストに関してはかなり見栄えを制御できます。しかし、強制的なフォントサイズや色の制御は、見る人の環境によってはかえって見づらくなる可能性もありますので、その使用に当たっては注意が必要です。できれば、FONT要素の使用は避け、サイズはBIG・SMALL要素、スタイルはTT・I・B・SUB・SUP要素あたりで調整するのが無難です。

B

<B>~</B>

ボールド（Bold text style）　　テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

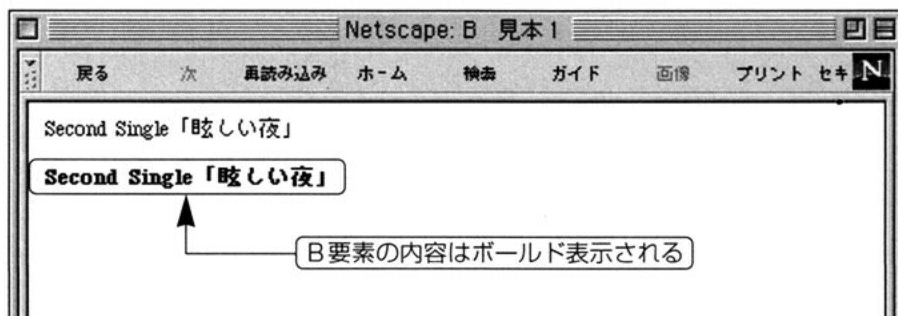
親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER
BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

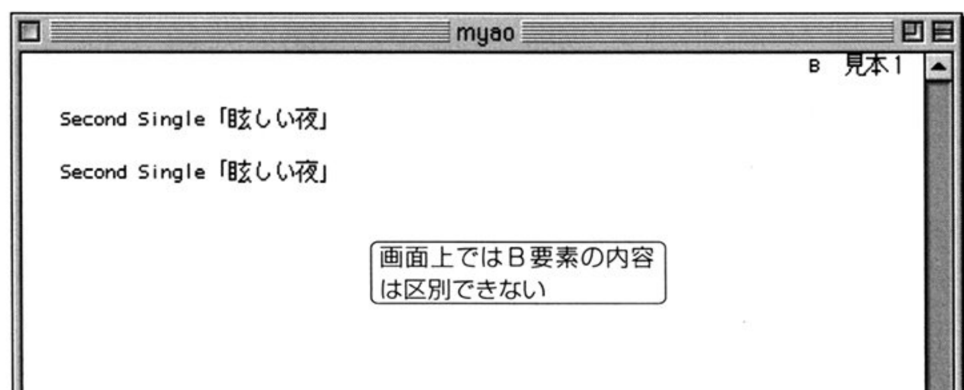
フォントスタイルを大まかに指定する要素です。

開始タグ<B>と終了タグ</B>でマークアップされたテキストは、通常、ボールド表示されます。

```
<P>Second Single「眩しい夜」</P>
<P><B>Second Single「眩しい夜」</B></P>
```



B要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

U

<U>~</U>

アンダーライン（Underlined text style）　テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

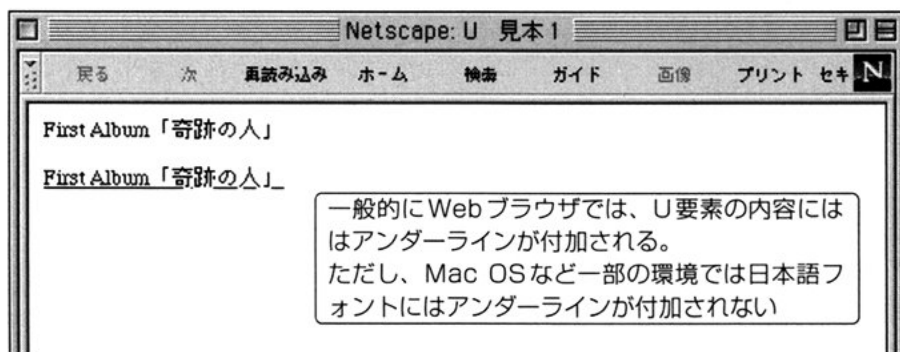
親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER
BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

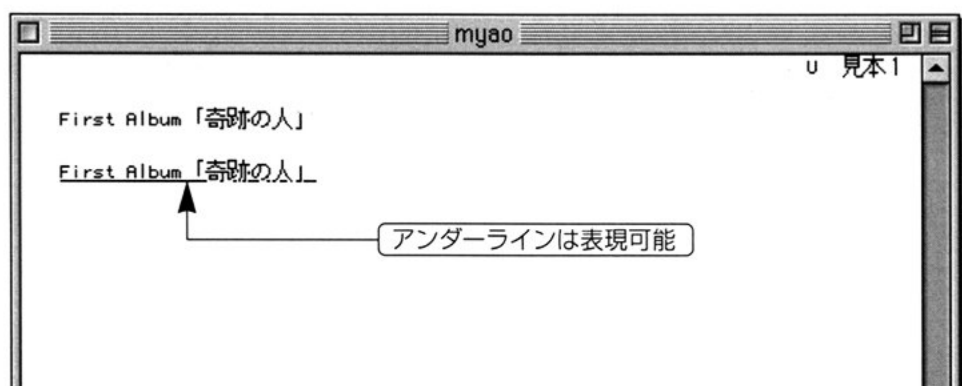
フォントスタイルを大まかに指定する要素です。

開始タグ<U>と終了タグ</U>でマークアップされたテキストは、アンダーライン表示されます。

```
<P>First Album「奇跡の人」</P>
<P><U>First Album「奇跡の人」</U></P>
```



U要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

NN 3 IE 3 NN 4 IE 4

STRIKE

<STRIKE>～</STRIKE>

取消線（STRIKE-through text style） **テキストレベル要素**

内容 ► TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

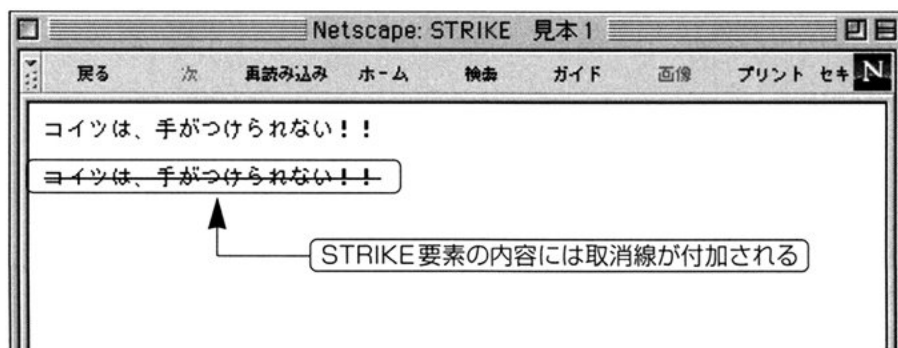
親要素 ► BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER
BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

フォントスタイルを大まかに指定する要素です。

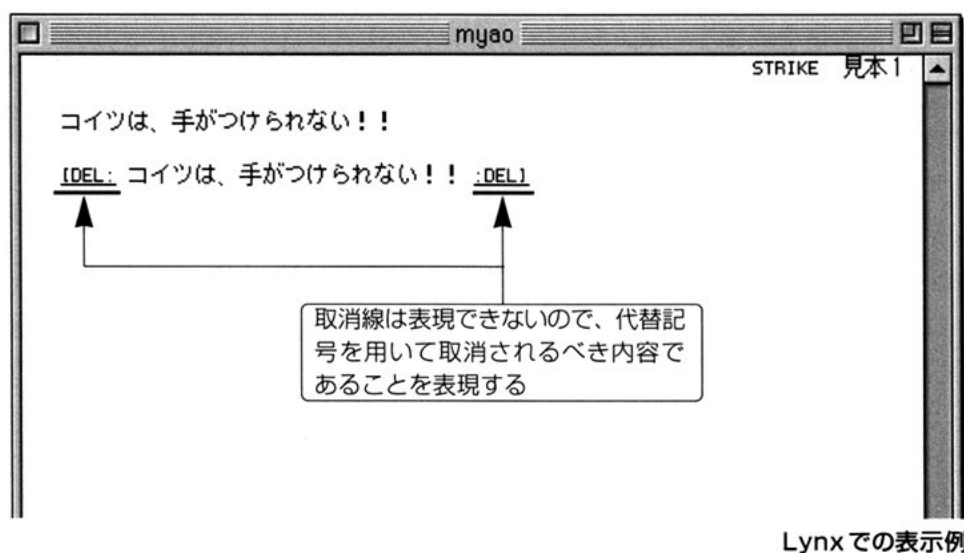
開始タグ<STRIKE>と終了タグ</STRIKE>でマークアップされたテキストは、取消線が付加されます。

```
<P>コイツは、手がつけられない！！</P>
<P><STRIKE>コイツは、手がつけられない！！</STRIKE></P>
```



STRIKE要素の使用例

フォント



Lynxでの表示例

なお、HTML 3.0の審議段階で提案されていた同様の機能を持つS要素は、HTML 3.2では採用されていません。

属性一覧

なし

属性解説

なし

NN 3 IE 3 NN 4 IE 4

BIG

<BIG>~</BIG>

大きめのフォント（places text in a large font）　テキストレベル要素

内容▶ TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
IMG　APPLET　FONT　BASEFONT　BR　MAP　INPUT
SELECT　TEXTAREA　SCRIPT
一般的なテキスト

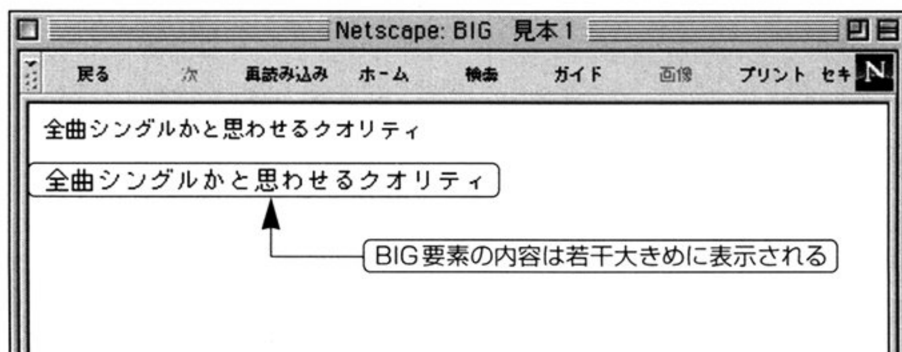
親要素▶ BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　DT　DD　LI　DIV　CENTER
BLOCKQUOTE　FORM　CAPTION　TH　TD
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
APPLET　FONT

要素の概要

フォントスタイルを大まかに指定する要素です。

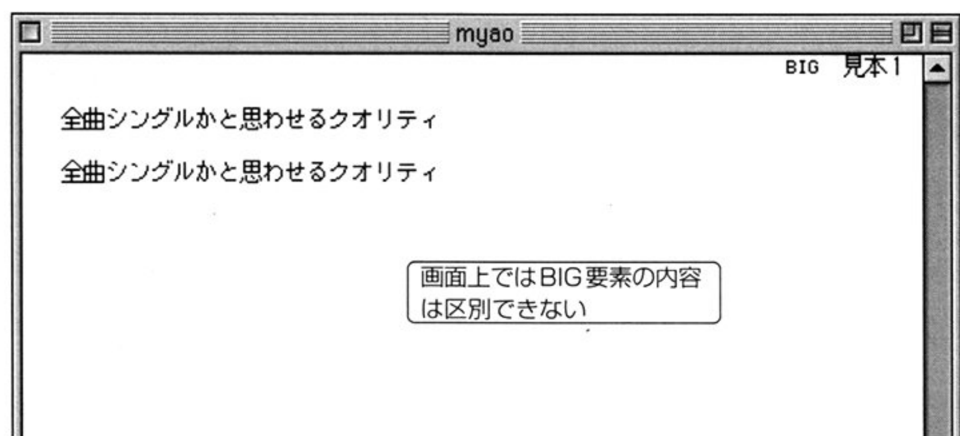
開始タグ<BIG>と終了タグ</BIG>でマークアップされたテキストは、設定されている標準フォントより若干大きめに表示されます。

```
<P>全曲シングルかと思わせるクオリティ</P>
<P><BIG>全曲シングルかと思わせるクオリティ</BIG></P>
```



BIG要素の使用例

フォント



Lynxでの表示例

属性一覧

なし

属性解説

なし

NN 3 IE 3 NN 4 IE 4

SMALL

<SMALL>~</SMALL>

小さめのフォント (places text in a small font)　テキストレベル要素

内容 TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

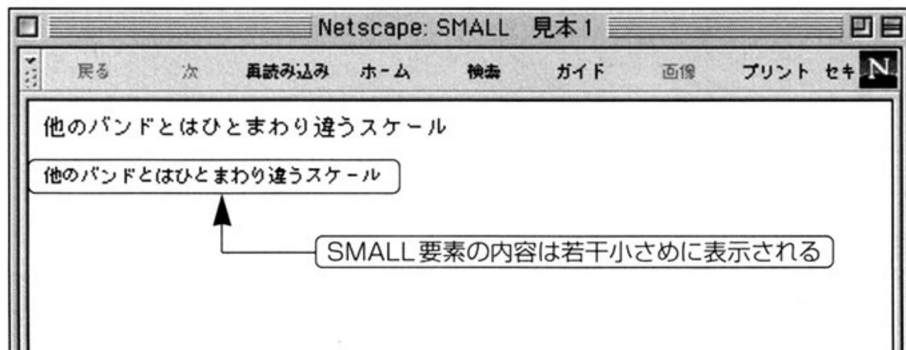
親要素 BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER
BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

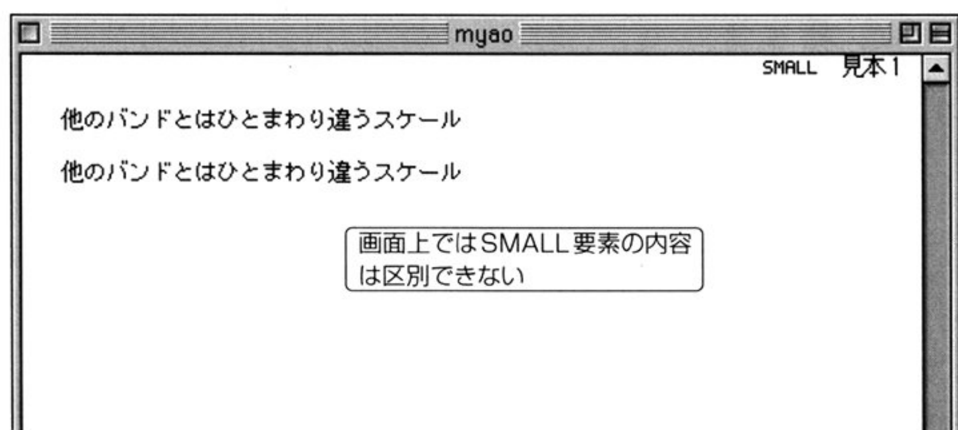
フォントスタイルを大まかに指定する要素です。

開始タグ<SMALL>と終了タグ</SMALL>でマークアップされたテキストは、通常、設定されている標準フォントより若干小さめに表示されます。

```
<P>他のバンドとはひとまわり違うスケール</P>
<P><SMALL>他のバンドとはひとまわり違うスケール</SMALL></P>
```



SMALL要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

SUB

_~

下付添字 (places text in a SUBscript style)　テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

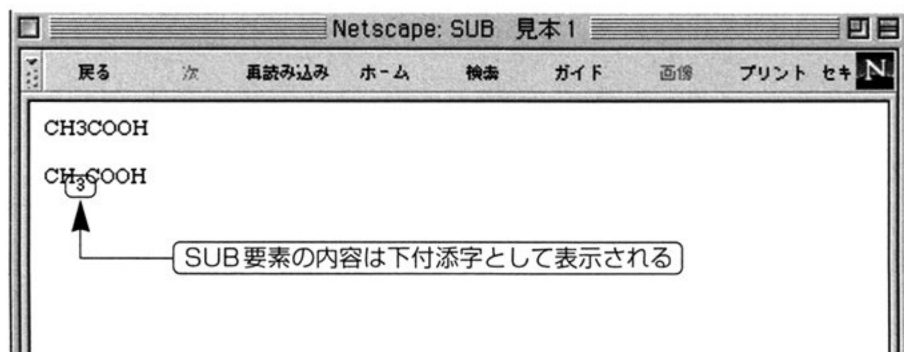
親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER
BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

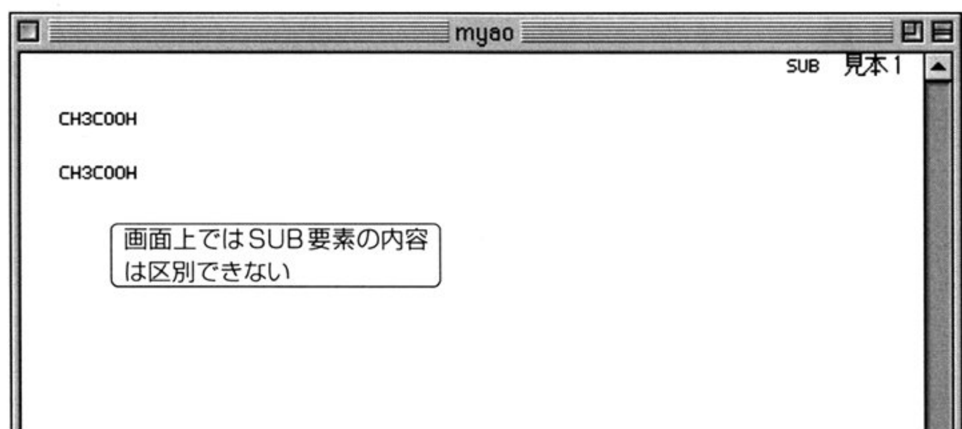
フォントスタイルを大まかに指定する要素です。

開始タグ_{と終了タグ}でマークアップされたテキストは、下付添字として表示されます。

```
<P>CH3COOH</P>
<P>CH<SUB>3</SUB>COOH</P>
```



SUB要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

SUP

[~]

上付添字（places text in a SUPerscript style）　テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

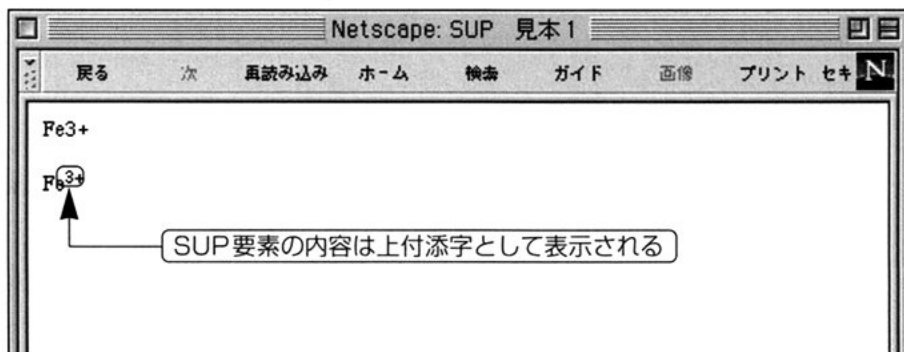
親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER
BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

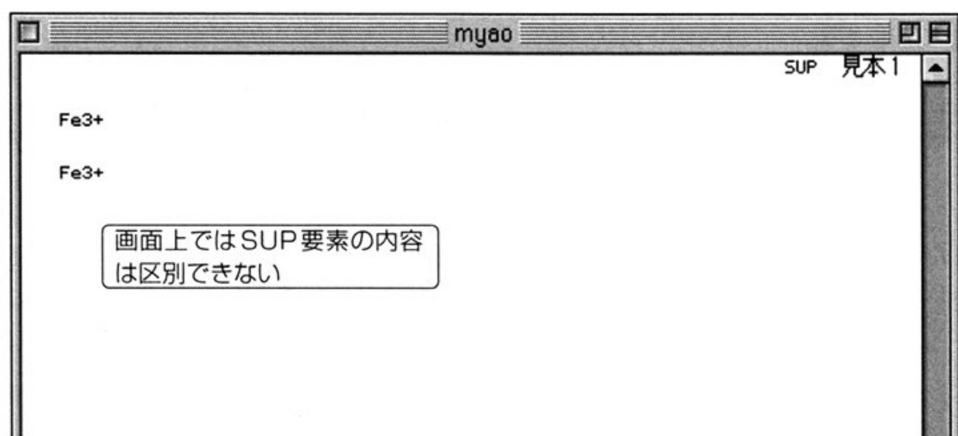
フォントスタイルを大まかに指定する要素です。

開始タグ^{と終了タグ}でマークアップされたテキストは、通常、上付添字として表示されます。

```
<P>Fe3+</P>
<P>Fe<SUP>3+</SUP></P>
```



SUP要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

EM

<EM>~</EM>

強調（basic EMphasis）　テキストレベル要素

内容 TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM　STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A　IMG　APPLET　FONT　BASEFONT　BR　MAP　INPUT　SELECT　TEXTAREA　SCRIPT
一般的なテキスト

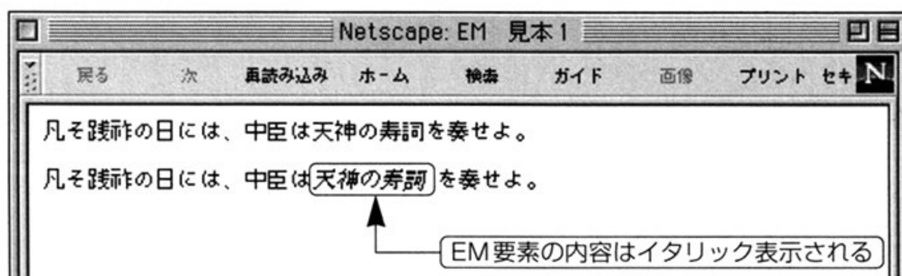
親要素 BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　LI　DT　DD　PRE　DIV　CENTER　BLOCKQUOTE　FORM　CAPTION　TH　TD
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM　STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A　APPLET　FONT

要素の概要

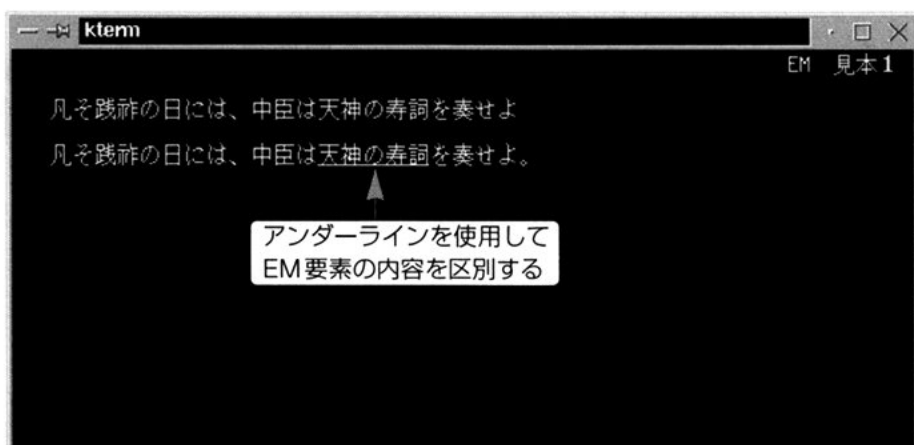
文章中の語句を強調したい場合に使用する要素です。

開始タグ<EM>と終了タグ</EM>でマークアップされた語句は、Webブラウザで表示すると周囲のテキストから区別できるようにイタリック表示されます。また、微妙にEM要素の前後に空白を挿入して目立たせるWebブラウザもあります（Netscape Navigatorなど）。

```
<P>凡そ践祚の日には、中臣は天神の寿詞を奏せよ。</P>
<P>凡そ践祚の日には、中臣は<EM>天神の寿詞</EM>を奏せよ。</P>
```



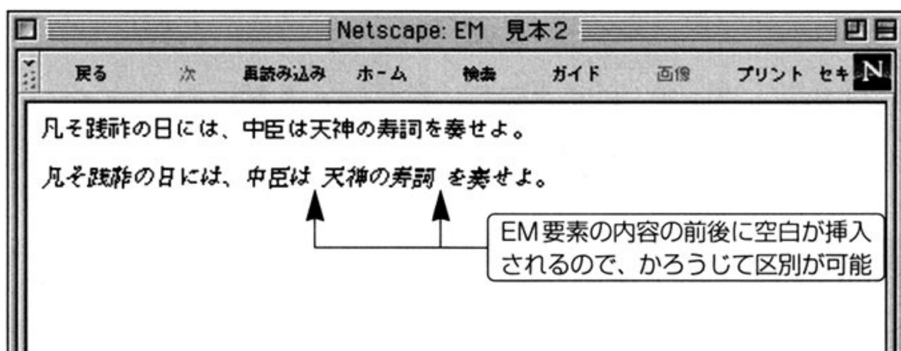
EM要素の使用例



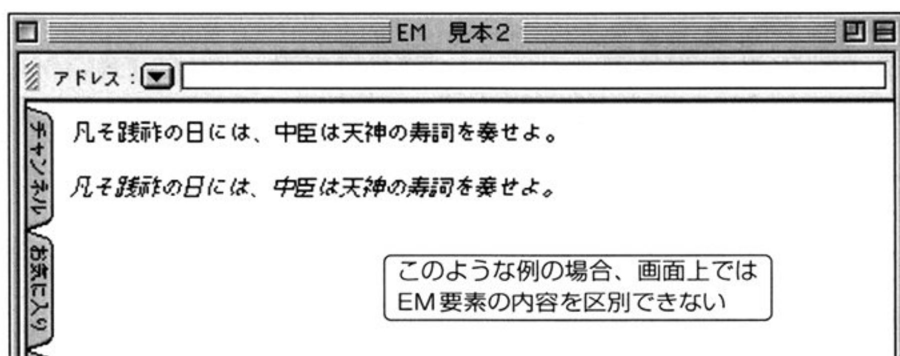
Lynxでの表示例

注意 EM要素は語句を強調するために、一般的にはイタリックで表示されますが、周囲の文章が既にI要素（イタリック）であれば、どう表現されるのでしょう？
EM要素の本来の意味からすれば、なんらかのフォントスタイル変更が自動的に行われなければならないはずです。しかし現在のところ、そこまで気の利いたWebブラウザはないようです。ただ、Netscape Navigatorなどのように、EM要素の前後に空白を挿入して目立たせているものでは、なんとか周囲のテキストと区別することが可能です（わかりやすいとは言えませんが）。

```
<P>凡そ践祚の日には、中臣は天神の寿詞を奏せよ。</P>
<P><I>凡そ践祚の日には、中臣は<EM>天神の寿詞</EM>を奏せよ。
</I></P>
```



Netscape Navigatorでの表示例



EM要素のInternet Explorerでの表示例

属性一覧

なし

属性解説

なし

Column　フレーズ要素とフォント関連の要素の違い

HTMLではテキスト中の特定の語句（フレーズ）の内容を意味づけするために「フレーズ要素（Phrase element）」と総称される要素が用意されています。たとえば、テキスト中にある人名や専門用語を強調したい場合、引用元文書のタイトル、プログラムのソースなどを明示したい場合などに使用されます。

Webブラウザでは、フレーズ要素を周囲のテキスト（いわゆる地の文）から区別するために、表示フォントスタイルや種類を変更したりします。そのため、使用方法がフォント関連の要素と混同されやすいのですが、フォント関連の要素が直接的に表示スタイルを指定するのに対して、フレーズ要素の表示スタイルは、あくまでも便宜的なものです。要は最低限周囲のテキストと区別できればいいのですから、Webブラウザや環境によって表示スタイルが異なるのも当然です。また、その区別は画面上とは限りません。たとえば、音声出力が可能なWebブラウザや専用アプリケーションでは、音量や音程、あるいは速度を変更して表現するかもしれません。見た目の機能は似ていても両要素の意味は全く異なります。くれぐれもフレーズ要素を「フォントスタイルの変更」のために使用しないよう、気をつけてください。

NN 3 IE 3 NN 4 IE 4

STRONG

<STRONG>~</STRONG>

より強い強調（STRONG emphasis）　テキストレベル要素

内容 TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

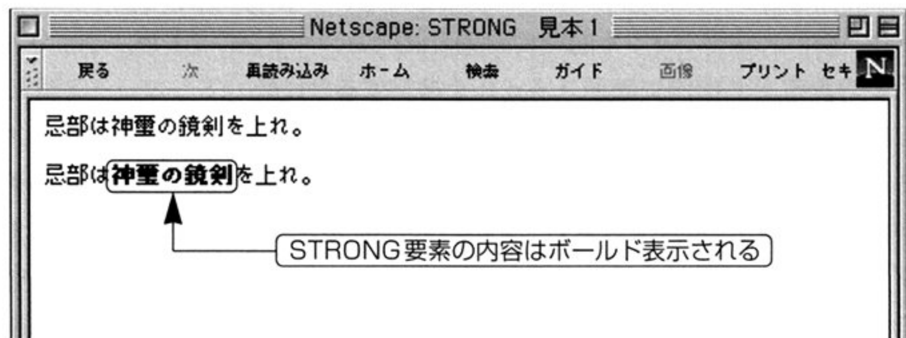
親要素 BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P LI DT DD PRE DIV CENTER BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

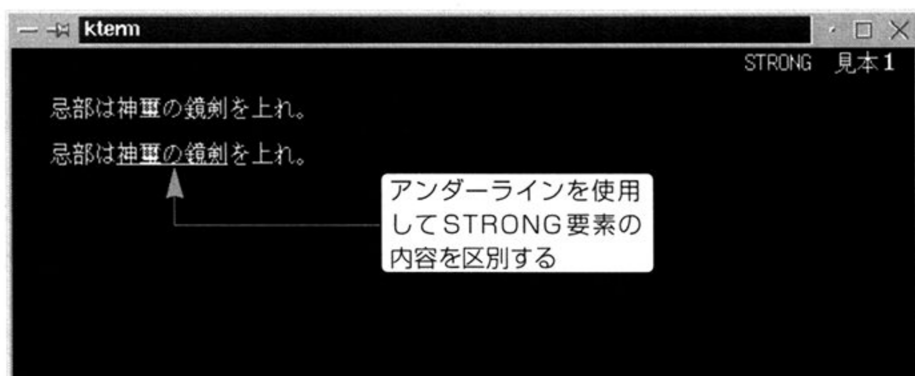
文章中の語句を並みの強調よりさらに強調したい場合に使用する要素です。

開始タグ<STRONG>と終了タグ</STRONG>でマークアップされた語句は、周囲のテキストから区別できるようにボールド表示されます。

```
<P>忌部は神璽の鏡剣を上れ。</P>
<P>忌部は<STRONG>神璽の鏡剣</STRONG>を上れ。</P>
```



STRONG要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

DFN

<DFN>～</DFN>

定義語句（DEfiNing instans of the enclosed term）　テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

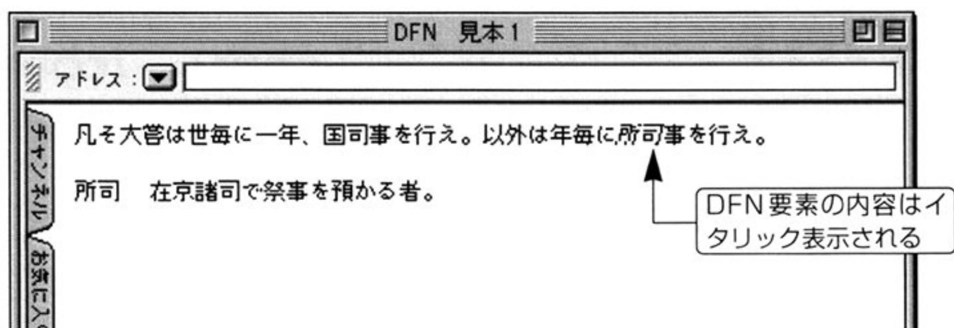
親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P LI DT DD PRE DIV CENTER BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

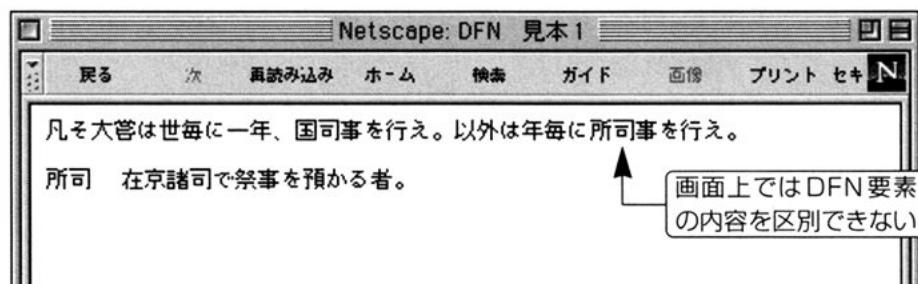
文章中の語句を定義された語句として強調したい場合に使用する要素です。

開始タグ<DFN>と終了タグ</DFN>でマークアップされた語句は、Internet Explorerでは表示する場合、周囲のテキストから区別できるようにイタリック表示されますが、Netscape Navigator（次ページ「注意」参照）やLynxでは表示上の変化はありません。

```
<P>凡そ大嘗は世毎に一年、国司事を行え。以外は年毎に<DFN>所司
</DFN>事を行え。</P>
<DL>
    <DT>所司</DT>
    <DD>在京諸司で祭事を預かる者。</DD>
</DL>
```



DFN要素のInternet Explorerでの表示例



DFN要素のNetscape Navigatorでの表示例

属性一覧

なし

属性解説

なし

Nn 3 | Ie 3 | Nn 4 | Ie 4

CODE

<CODE>～</CODE>

プログラムコード（program CODE）　テキストレベル要素

内容 TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

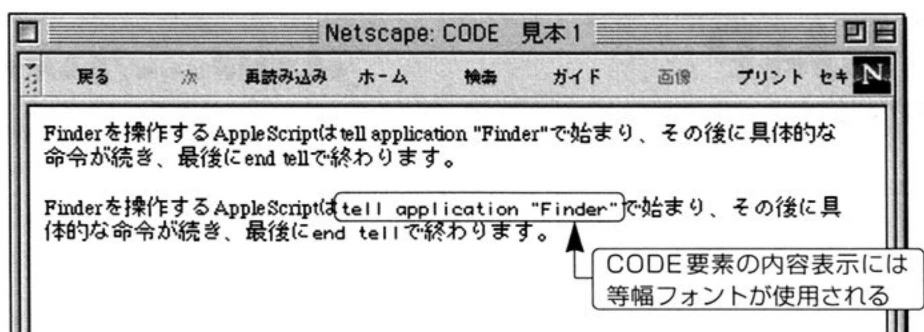
親要素 BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P LI DT DD PRE DIV CENTER BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

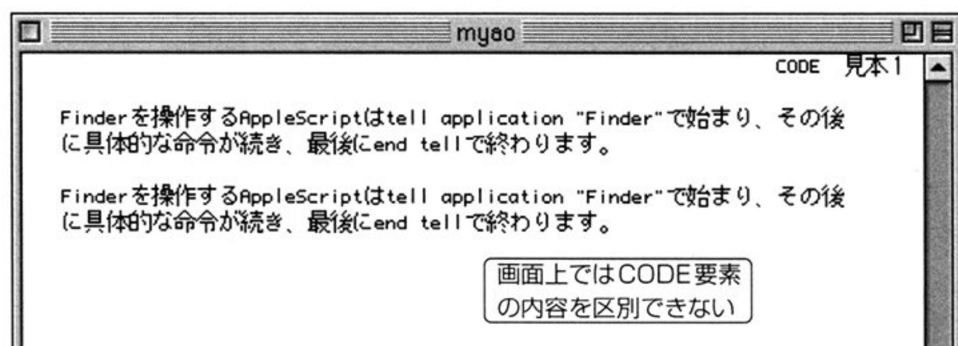
文章中の語句が、コンピュータのプログラムコードからの抜粋であることを強調したい場合に使用する要素です。

開始タグ<CODE>と終了タグ</CODE>でマークアップされた語句は、周囲のテキストから区別できるよう、表示に等幅フォントが使用されます。

```
<P>Finderを操作するAppleScriptはtell application “Finder”で始まり、その後に具体的な命令が続き、最後にend tellで終わります。
</P>
<P>Finderを操作するAppleScriptは<CODE>tell application
“Finder”</CODE>で始まり、その後に具体的な命令が続き、最後に
<CODE>end tell</CODE>で終わります。</P>
```



CODE要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

SAMP

<SAMP>~</SAMP>

サンプル出力（SAMPle output from programs and script）　テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

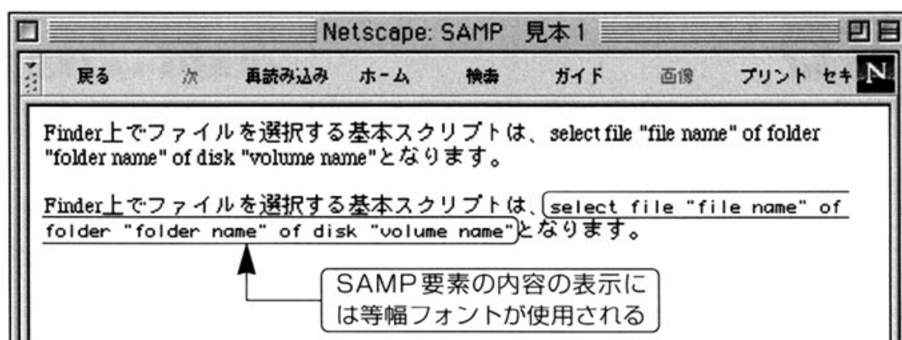
親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P LI DT DD PRE DIV CENTER BLOCKQUOTE FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

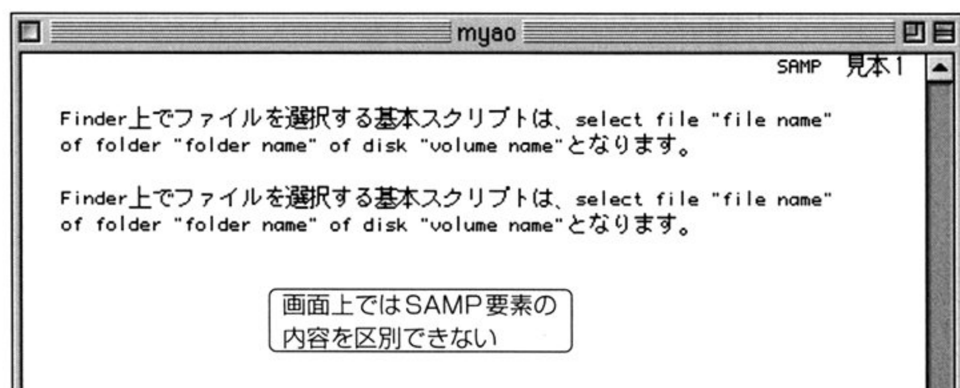
文章中の語句が、コンピュータのプログラムやスクリプトからのサンプル出力であることを強調したい場合に使用する要素です。

開始タグ<SAMP>と終了タグ</SAMP>でマークアップされた語句は、周囲のテキストから区別できるよう、表示に等幅フォントが使用されます。

```
<P>Finder上でファイルを選択する基本スクリプトは、select file
"file name" of folder "folder name" of disk "volume
name"となります。</P>
<P>Finder上でファイルを選択する基本スクリプトは、<SAMP>select
file "file name" of folder "folder name" of disk
"volume name"</SAMP>となります。</P>
```



SAMP要素の表示例



Lynxでの表示例

属性一覧

なし

属性解説

なし

KBD

<KBD>~</KBD>

ユーザによる入力（text to be typed by the user）　テキストレベル要素

内容 ▶ TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
IMG　APPLET　FONT　BASEFONT　BR　MAP　INPUT
SELECT　TEXTAREA　SCRIPT
一般的なテキスト

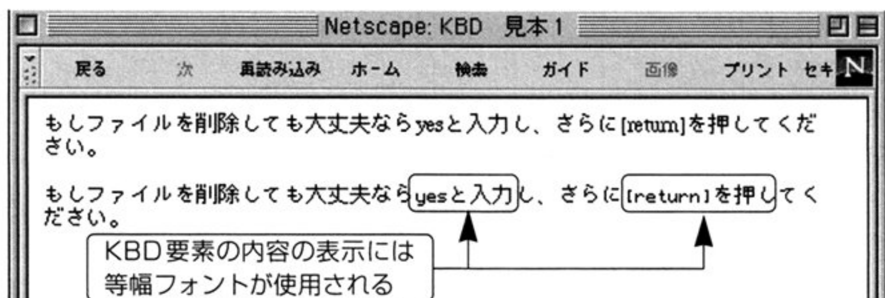
親要素 ▶ BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　LI　DT　DD　PRE　DIV　CENTER　BLOCKQUOTE
FORM　CAPTION　TH　TD
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
APPLET　FONT

要素の概要

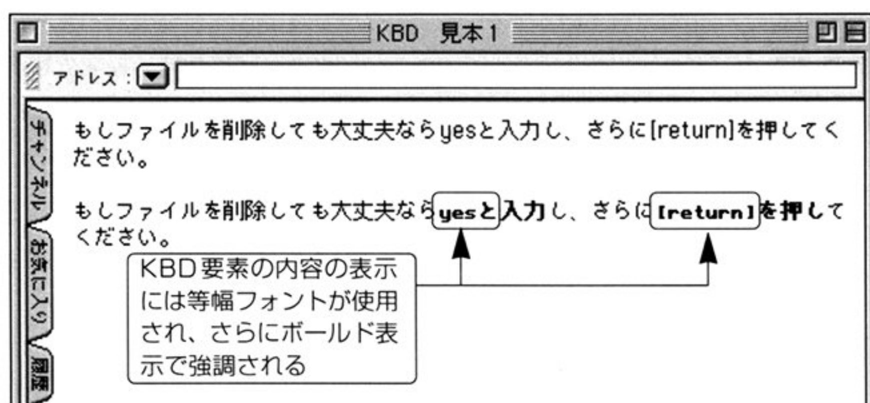
文章中の語句が、ユーザの操作によって入力されるべきものであることを強調したい場合に使用する要素です。

開始タグ<KBD>と終了タグ</KBD>でマークアップされた語句は、周囲のテキストから区別できるよう、表示に等幅フォントが使用されます。

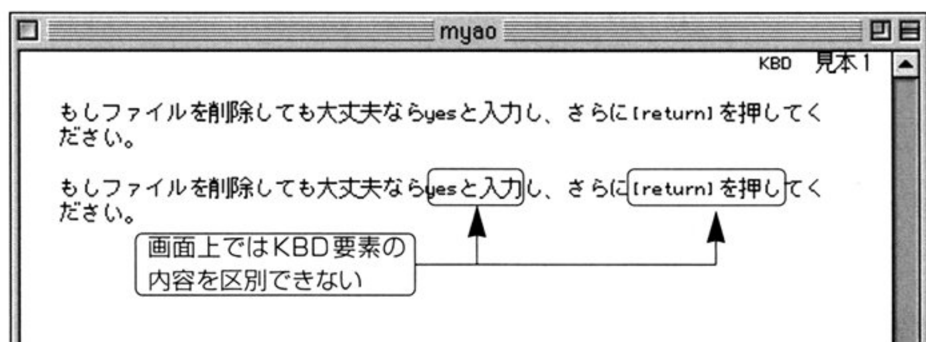
```
<P>もしファイルを削除しても大丈夫ならyesと入力し、さらに[return]を
押してください。</P>
<P>もしファイルを削除しても大丈夫なら<KBD>yesと入力</KBD>し、さら
に<KBD>[return]を押し</KBD>てください。</P>
```



KBD要素のNetscape Navigatorでの表示例



KBD要素のInternet Explorerでの表示例



Lynxでの表示例

属性一覧

なし

属性解説

なし

VAR

<VAR>~</VAR>

変数や引数（VARiables or arguments to commands）　テキストレベル要素

内容 TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
IMG　APPLET　FONT　BASEFONT　BR　MAP　INPUT
SELECT　TEXTAREA　SCRIPT
一般的なテキスト

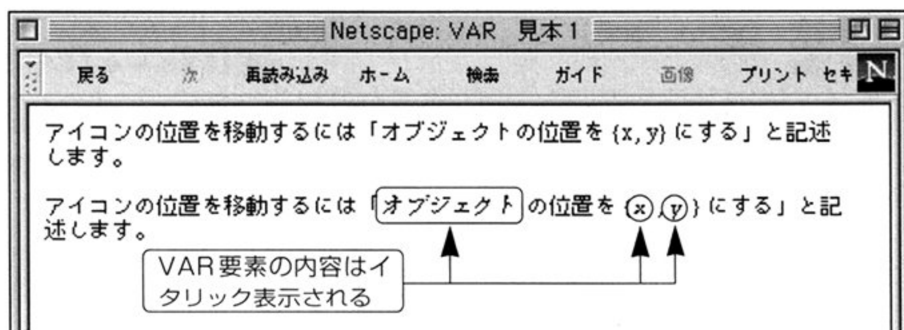
親要素 BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　LI　DT　DD　PRE　DIV　CENTER　BLOCKQUOTE
FORM　CAPTION　TH　TD
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
APPLET　FONT

要素の概要

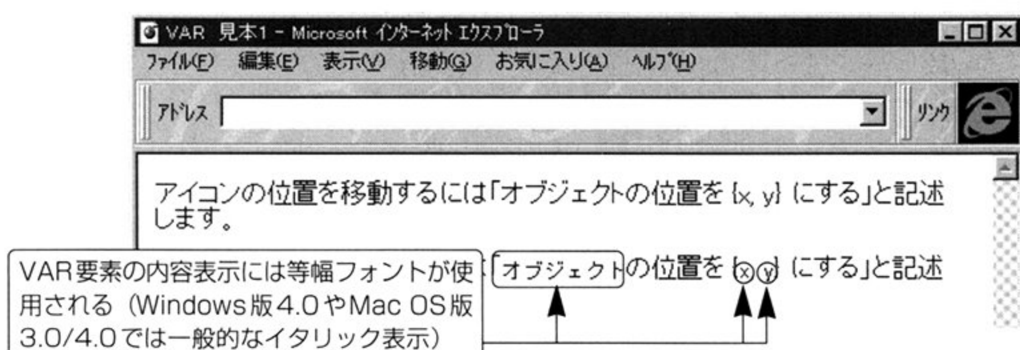
文章中の語句が、コンピュータのプログラムやスクリプトの変数や引数であることを強調したい場合に使用する要素です。

開始タグ<VAR>と終了タグ</VAR>でマークアップされた語句は、周囲のテキストから区別できるようイタリック表示されます。

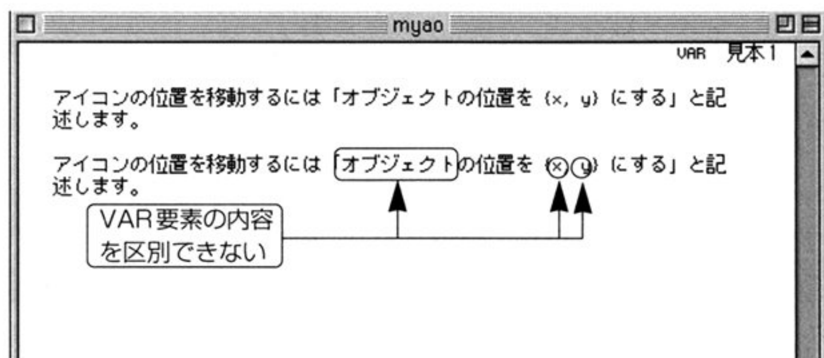
```
<P>アイコンの位置を移動するには「オブジェクトの位置を {x, y} にする」
と記述します。</P>
<P>アイコンの位置を移動するには「<VAR>オブジェクト</VAR>の位置を
{<VAR>x</VAR>, <VAR>y</VAR>} にする」と記述します。</P>
```



VAR要素の使用例



VAR要素のWindows版Internet Explorer 3.0での表示例



Lynxでの表示例

属性一覧

なし

属性解説

なし

NN 3 IE 3 NN 4 IE 4

CITE

<CITE>~</CITE>

引用や参照語句（CITarions or references to other sources）　テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
IMG APPLET FONT BASEFONT BR MAP INPUT
SELECT TEXTAREA SCRIPT
一般的なテキスト

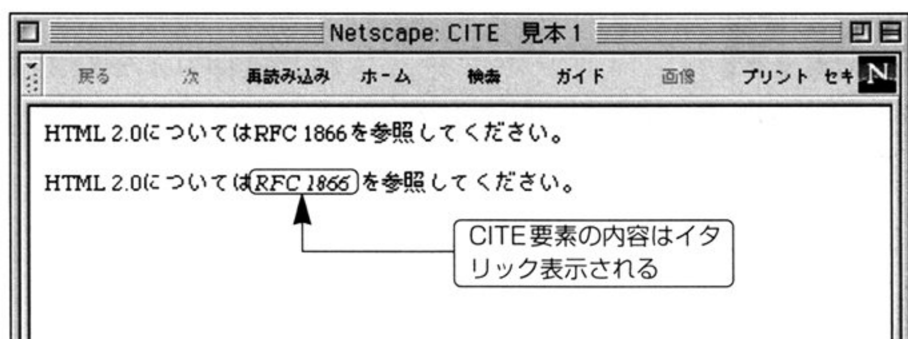
親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P LI DT DD PRE DIV CENTER BLOCKQUOTE
FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
APPLET FONT

要素の概要

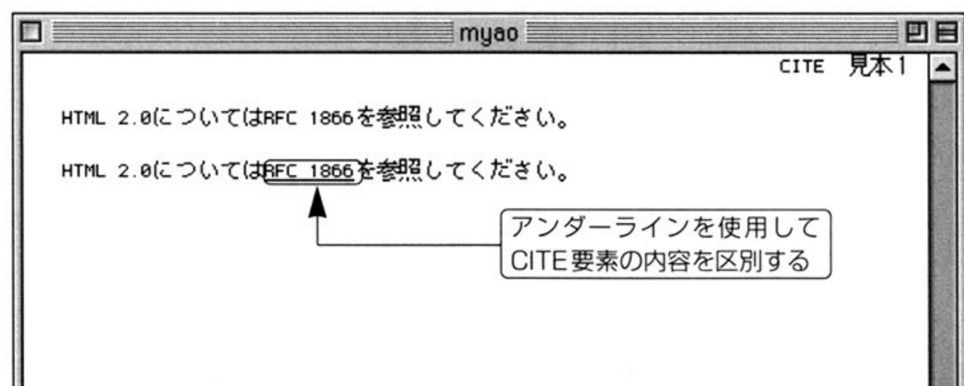
引用、参照元文書のタイトル・著者名などを明示する場合に使用する要素です。

開始タグ<CITE>と終了タグ</CITE>でマークアップされた語句は、周囲のテキストから区別できるようイタリック表示されます。

```
<P>HTML 2.0についてはRFC 1866を参照してください。</P>
<P>HTML 2.0については<CITE>RFC 1866</CITE>を参照してください。
</P>
```



CITE要素の使用例



Lynxでの表示例

属性一覧

なし

属性解説

なし

NN 3 IE 3 NN 4 IE 4

FORM

<FORM>~</FORM>

フォーム（The FORM element） ブロックレベル要素

内容 ▶ H1 H2 H3 H4 H5 H6
ADDRESS
P UL OL DL (DIR) (MENU) PRE DIV
CENTER BLOCKQUOTE ISINDEX HR TABLE
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 ▶ BODY
DIV CENTER BLOCKQUOTE FORM TH TD DD
LI

要素の概要

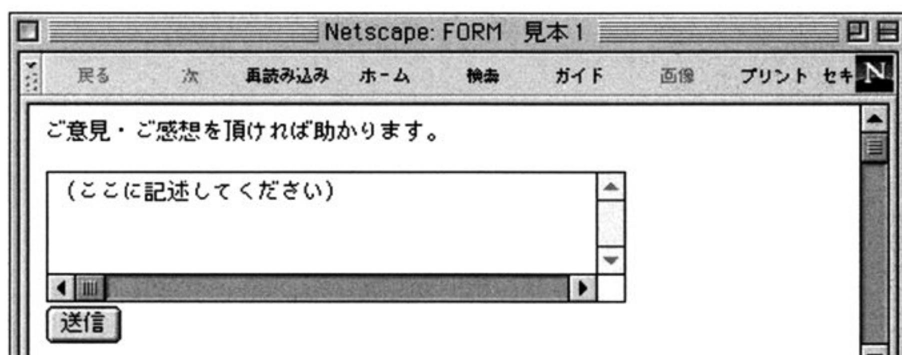
アンケートやメッセージ（ゲストブックや伝言板など）を受け取るために、多くのWebサイトで「フォーム」が使用されています。このフォームはWebブラウザだけでなく、若干の機能制限はあるもののLynxなどテキストブラウザでも使用可能です。FORM要素は、このフォーム機能を実現するための要素です。

開始タグ<FORM>と終了タグ</FORM>の間には、フォームに必要なさまざまな要素（見出し・アドレス・ブロックレベル・テキストレベルの各要素と一般的なテキスト）を置くことができますが（次ページ「注意」参照）、メインとなるのは具体的なフォーム機能の設定に関わるINPUT・SELECT・TEXTAREA要素です。

なお、フォーム機能ではデータをメールとして送信することも可能ですが、これはWebブラウザ側の機能によって制限が起こったり後処理が面倒になったりします。そのため、一般的にはデータをサーバのCGIスクリプトに渡して処理します。本書では属性の機能などを説明するために、メールソフトで受信した結果を載せていますが、これは一般的な使用例でないことをあらかじめお断りしておきます。CGIスクリプトについては、専門の解説書をご覧ください。

注意 HTML 3.2の仕様書には「FORM要素の内容は、INPUT・SELECT・TEXTAREA要素のみ」と書かれていますが、DTDでは「FORM要素の内容は、見出し・アドレス・ブロックレベル・テキストレベルの各要素と一般的なテキスト」と定義されていますので、本書の解説はDTDの定義に従いました。
なお、実はINPUT・SELECT・TEXTAREA要素をFORM要素外に置くこともできます。HTML 3.2では具体的な使用方法がなさそうですが、HTML 4.0ではスクリプトと併用して、フォームデータの送信以外のユーザーインターフェイスとして機能させることができます。

```
<P>ご意見・ご感想を頂ければ助かります。</P>
<FORM ACTION="http://www.heian.go.jp/iken.cgi"
METHOD="POST">
<TEXTAREA NAME="iken" ROWS="3" COLS="40">（ここに記述して
ください）</TEXTAREA><BR>
<INPUT TYPE="submit" VALUE="送信">
</FORM>
```



簡単なフォームの例

属性一覧

●ACTION

値	働き	H3	N4	I4
url	送信したデータを処理するCGIスクリプトのURL	○	○	○

属性は省略可

●METHOD（初期値：GET）

値	働き	H3	N4	I4
GET	データをACTION属性で設定されたURLの後に追加して送信	○	○	○
POST	データのみを送信	○	○	○

属性は省略可

●ENCTYPE（初期値：application/x-www-form-urlencoded）

		H3	N4	I4
値	application/x-www-form-urlencoded	○	○	○
働き	「Content-Type」フィールドを付加せず送信			
値	multipart/form-data	○	○	○
働き	「Content-Type」フィールドを付加して送信			

属性は省略可

変数　*url*=URL

属性解説

●ACTION属性

フォーム機能によって送信されたデータを処理するCGIスクリプトのURLを設定します。

```
<FORM ACTION="http://www.heian.go.jp/iken.cgi">
```

上記の例では、送信したデータは「http://www.heian.go.jp/iken.cgi」というCGIスクリプトによって処理されます。

また、「mailto:メールアドレス」を設定すると、データはメールとして設定したメールアドレスへ送信されます。

```
<FORM ACTION="mailto:nkoiyohar@heian.go.jp">
```

ただし、メール送信機能に対応していない（あるいはメールソフトを連動させることができない）Webブラウザでは、「mailto:メールアドレス」はうまく利用できません。

●METHOD属性

フォーム機能によるデータの送信方法を指定します。具体的な方法には「GET」と「POST」の2種類があり、属性を省略した場合は、GETが初期値として使用されます。

GETでは、データはACTION属性で設定されURLの後に追加して送信されます。

```
<P>お名前をお教えください。</P>
<FORM ACTION="http://www.heian.go.jp/namae.cgi"
METHOD="GET">
<INPUT TYPE="text" NAME="namae">
<INPUT TYPE="submit" VALUE="送信">
</FORM>
```



METHOD属性の使用例

「Masato Kamata」と入力して送信すると、「http://www.heian.go.jp/namae.cgi?namae=Masato+Kamata」の文字列がサーバへ送信される。

?	：URLとのデータの区切り記号
namae	：INPUT要素のNAME属性で設定したデータフィールド名
Masato+Kamata	：データフィールド「namae」の値。空白文字は「+」で置き換えられる

一方POSTでは、データ部分だけが送信されます。

このGETとPOSTの使い分けですが、一般的には受信後のデータを特に加工・整理する必要のないもの、たとえばデータベースサービスの検索文字列の送信などにGETが使用されます。また、ENCTYPE属性の値は「application/x-www-form-urlencoded」しか使用できないので、日本語などを大量に送信するのには向いていません。

反対に、データを受信後加工するなどの作業が必要なもの、たとえばゲストブックやアンケートなどではPOST が使用されています。また、長めのデータを送信する場合は、POSTの方が安全といわれていますし、送信するデータをWebブラウザのURL欄に表示したくない場合も、POSTを使用します。ただ、サーバによってはPOSTではうまく動作しない場合もあるそうです。

●ENCTYPE属性

送信するデータのMIMEのContent-Type (P.59参照) を設定します。Webブラウザ(あるいはメールソフトなどのアプリケーション)は、データのヘッダのContent-Typeフィールドにこの属性値を入れて送信します。

1. ENCTYPE="application/x-www-form-urlencoded"の場合

ENCTYPE属性の初期値です。

ASCIIの文字セット (P.309参照) に含まれない文字は、「%＋16進数×２」の組み合わせにエンコード(変換) して送信されます。また、空白文字は「+」に置き換えられます。さらにフィールドが複数ある場合は、区切りに「&」記号が用いられます。また、データの受信側は、一般的にCGIスクリプトで解析して、いったん特定の符号化方式(UNIX上ではEUCが一般的) に復元して処理を行います。

2. ENCTYPE="multipart/form-data"の場合

値に「multipart/form-data」を使用すると、データは複数のパートからなるデータとして送信されます。仕組みはMIMEによるメールの添付ファイルと同じで、フォームの各フィールドが1つの添付ファイル(パート) として扱われます。また、各フィールドのデータの先頭にはContent-Dispositionフィールドが置かれ、データの内容がフォームのデータ (form-data) であることやフォームのフィールド名(INPUT・TEXTAREA要素のNAME属性の値) などが入れられます。

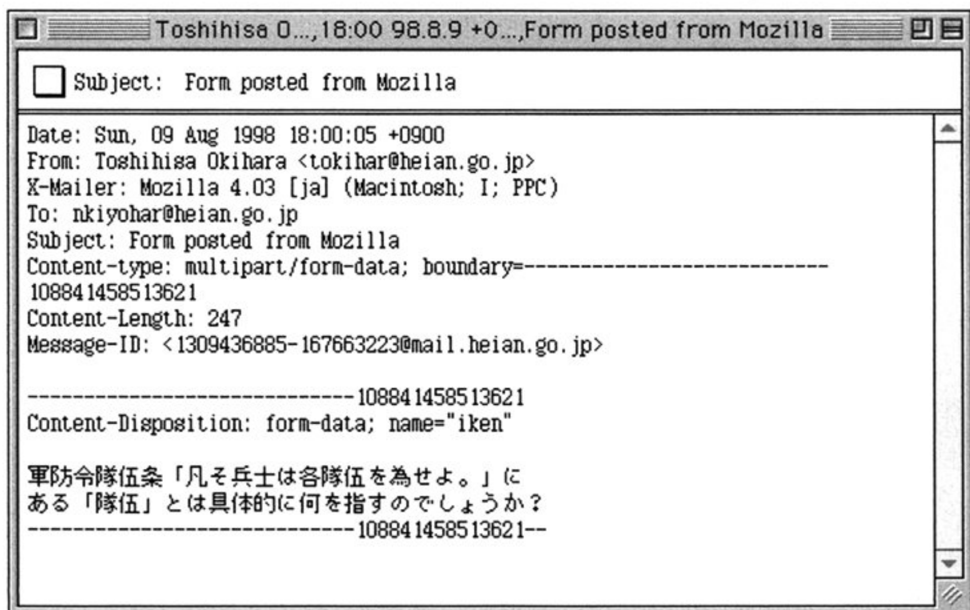
さらに、Content-Typeフィールドにcharsetパラメータが付加され、そこへ文字の符号化方式も入れられます。受信側のソフトウェアは、このcharsetパラメータ値によって、データをあれこれ解析することなく、オリジナルの符号化方式に合わせてデータを復元できます。そのため、この「multipart/form-data」は、ACTION属性で「mailto:メールアドレス」を設定し、データをメールとして受信する場合などによく使用されます。

ただし、これらの機能を有効に使用するには、送信側のWebブラウザ(あるいはメールソフトなどアプリケーション) 側がMIMEに基づくデータ送信に対応している必要があります。

```
<P>ご意見・ご感想を頂ければ助かります。</P>
<FORM ACTION="mailto:nkiyohar@heian.go.jp"
METHOD="POST" ENCTYPE="multipart/form-data">
<TEXTAREA NAME="iken" ROWS="3" COLS="40">（ここに記述して
ください）</TEXTAREA><BR>
<INPUT TYPE="submit" VALUE="送信">
</FORM>
```

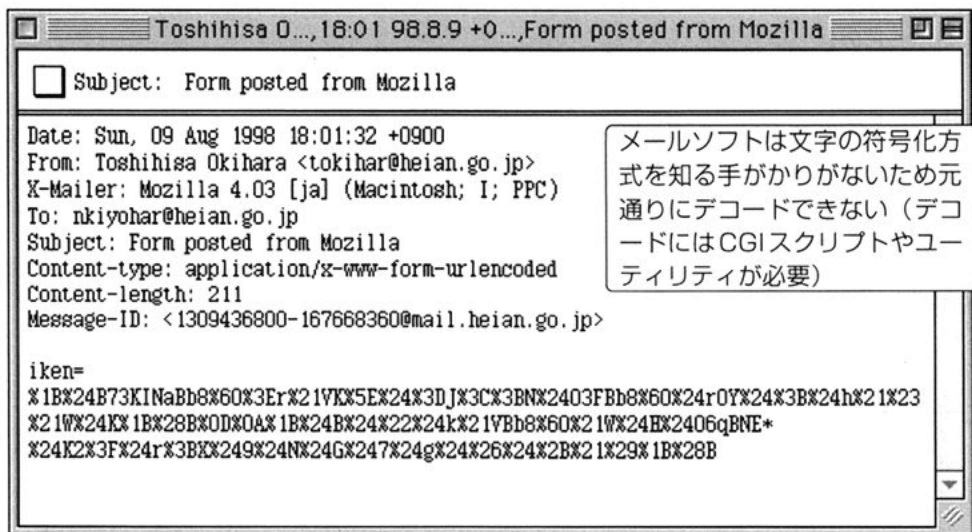


このテキストをメールとして送信



ENCTYPE="multipart/form-data"で送信されたメールの例

フォーム



ENCTYPE="application/x-www-form-urlencoded"で送信されたメールの例

NN 3 IE 3 NN 4 IE 4

INPUT

<INPUT>（終了タグなし）

入力フィールド・ボタンなど（The INPUT element）　テキストレベル要素

内容 ▶ 空

親要素 ▶ BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　PRE　DIV　CENTER　BLOCKQUOTE　FORM　CAPTION　TH　TD　DT　DD　LI
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM　STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A　APPLET　FONT

要素の概要

INPUT要素は、フォームのテキスト入力欄や選択ボタン、送信ボタンなどの設定に使用します。通常はFORM要素内に置かれますが、テキストレベル要素なので直接的にはさまざまな要素内に置くことができます。

なお、INPUTは開始タグ<INPUT>のみでマークアップされる、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

```
<P>お名前・ご連絡先をお教えください。</P>
<FORM ACTION="http://www.heian.go.jp/namae.cgi"
METHOD="POST">
  <P>
    名　前：<INPUT TYPE="text" NAME="namae"><BR>
    連絡先：<INPUT TYPE="text" NAME="address"><BR>
            <INPUT TYPE="submit" VALUE="送信">  <INPUT
TYPE="reset" VALUE="取消">
  </P>
</FORM>
```

他の要素の中に置くことができる

Netscape: INPUT 見本1

戻る 次 再読み込み ホーム 検索 ガイド 画像 プリント セキ

お名前・ご連絡先をお教えください。

名　前：

連絡先：

送信　取消

INPUT要素の使用例

属性一覧

●TYPE（初期値：text）

値	働き	H3	N4	I4
text	1行テキスト入力	○	○	○
password	パスワード入力	○	○	○
checkbox	チェックボックス（複数選択）	○	○	○
radio	ラジオボタン（単数選択）	○	○	○
submit	送信ボタン	○	○	○
image	画像付き送信ボタン（送信ボタンに画像を使用）	○	○	○
reset	入力データのリセットボタン	○	○	○
file	ファイル名入力	○	○	○
hidden	不可視フィールド	○	○	○

属性は省略可

●NAME

値	働き	H3	N4	I4
text	データフィールド名の設定	○	○	○

属性は省略可（ただし、その他の属性の設定による）

●VALUE

値	働き	H3	N4	I4
text	データの初期値や送信や取消ボタンのラベルの設定	○	○	○

属性は省略可

●CHECKED（論理型）

値	働き	H3	N4	I4
checked	チェックボックスやラジオボタンの初期値をオン	○	○	○

属性は省略可

●SIZE

値	働き	H3	N4	I4
number	テキスト入力欄の幅の設定（文字数）	○	○	○

属性は省略可

●MAXLENGTH

値	働き	H3	N4	I4
number	テキスト入力の最大文字数の設定	○	○	○

属性は省略可

●SRC

値	働き	H3	N4	I4
url	画像付き送信ボタンの画像データのURL	○	○	○

属性は省略可

●ALIGN

値	働き	H3	N4	I4
top	画像付き送信ボタンを上揃え	○	○	○
middle	画像付き送信ボタンを中揃え	○	○	○
bottom	画像付き送信ボタンを下揃え	○	○	○
left	画像付き送信ボタンを左揃え	○	○	○
right	画像付き送信ボタンを右揃え	○	○	○

属性は省略可

変数 *text*=一般的なテキスト　*number*=数値　*url*=URL

属性解説

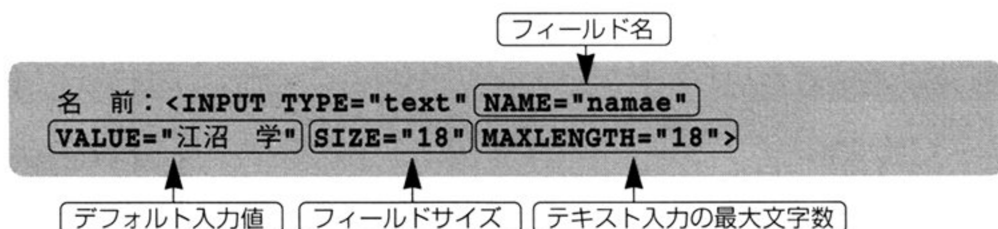
●TYPE属性

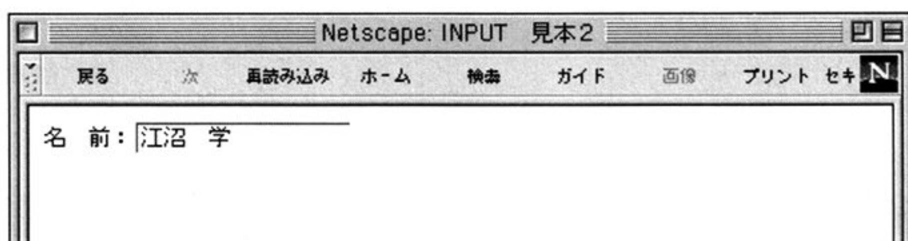
TYPE属性はデータ入力形式を指定します。デフォルトで入力される値やフィールドサイズなど詳細は、その他の属性で指定します。

1．TYPE="text"

textは1行のテキスト入力フィールドを表示します。

NAME属性は必須、その他詳細設定には、VALUE・SIZE・MAXLENGTH属性が使用されます。



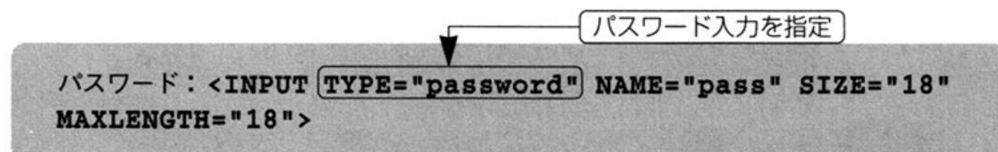


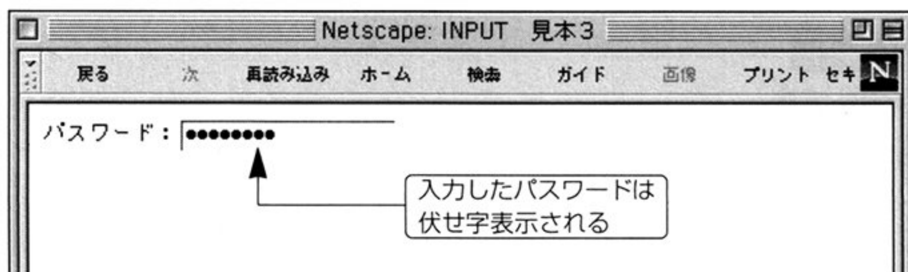
データ入力形式をテキストにした例

2．TYPE="password"

passwordはパスワード入力に使用します。入力データは伏せ字で表示されます。

NAME属性は必須、その他詳細設定には、VALUE・SIZE・MAXLENGTH属性が使用されます。





データ入力をパスワードにした例

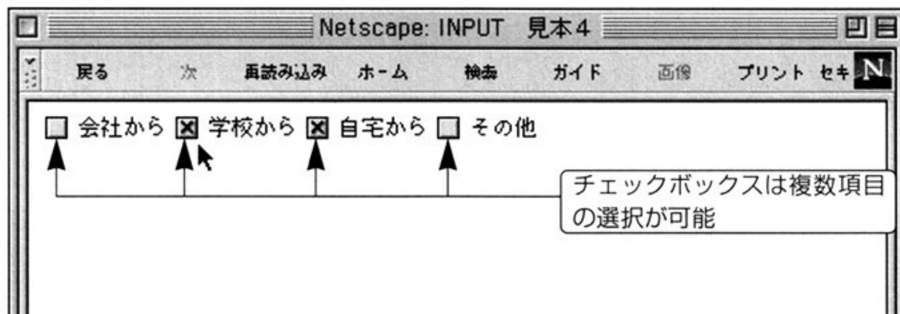
3. TYPE="checkbox"

checkboxはチェックボックスを表示します。チェックボックスは、複数の項目を選択する場合に使用されます。

NAME属性は必須、その他詳細設定には、VALUE・CHECKED属性が使用されます。

チェックボックスの指定

```
<INPUT TYPE="checkbox" NAME="place" VALUE="会社">
会社から
<INPUT TYPE="checkbox" NAME="place" VALUE="学校">
学校から
<INPUT TYPE="checkbox" NAME="place" VALUE="自宅">
自宅から
<INPUT TYPE="checkbox" NAME="place" VALUE="その他">
その他
```



データ入力をチェックボックスにした例

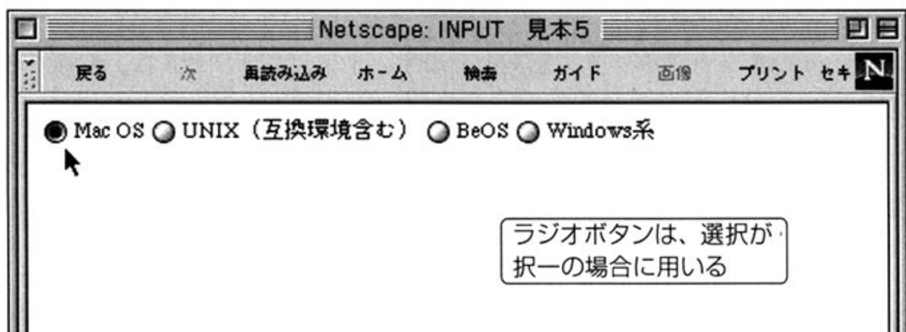
4. TYPE="radio"

radioはラジオボタンを表示します。ラジオボタンは、通常単数の項目を選択する場合に使用されます。

NAME属性は必須、その他詳細設定には、VALUE・CHECKED属性が使用されます。

ラジオボタンの指定

```
<INPUT TYPE="radio" NAME="os" VALUE="Mac OS">Mac OS
<INPUT TYPE="radio" NAME="os" VALUE="UNIX">UNIX（互換環
境含む）
<INPUT TYPE="radio" NAME="os" VALUE="BeOS">BeOS
<INPUT TYPE="radio" NAME="os" VALUE="Windows">Windows
```



データ入力をラジオボタンにした例

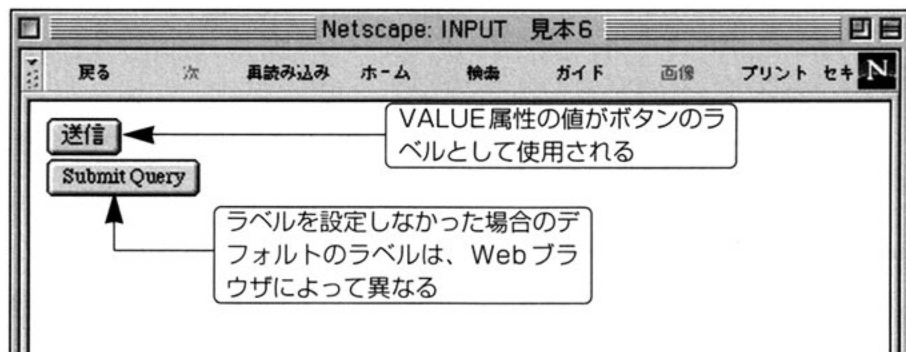
5．TYPE="submit"

submitは送信ボタンを表示します。送信ボタンをクリック（あるいはキー操作など）すると、FORM要素の設定に基づいて、入力データが送信されます。

その他詳細設定には、NAME・VALUE属性が使用されます。

```
<INPUT TYPE="submit" VALUE="送信"><BR>
<INPUT TYPE="submit">
```

ボタンのラベルはVALUE属性で設定する



ボタンの指定例

6．TYPE="image"

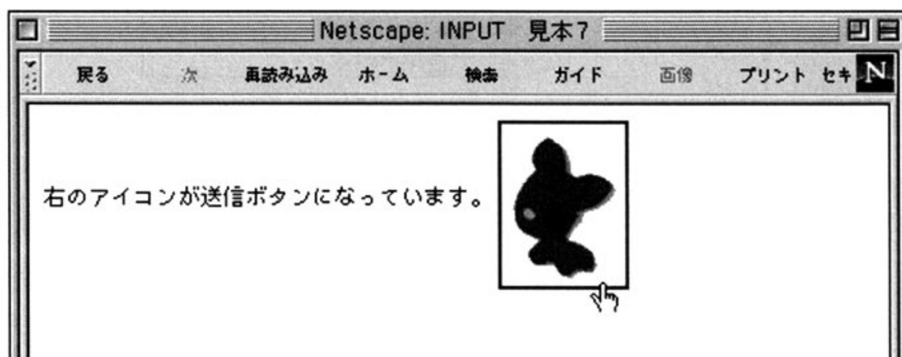
imageは送信ボタンのラベルに画像を使用する、画像付き送信ボタンを指定します。画像がマウスなどでクリックされた場合、データと共にそのクリックされた座標（画像の左端から数えたpixel数がx、上端から数えたpixel数がy）も送信されます。

その他詳細設定には、NAME・VALUE・SRC・ALIGN属性が使用されます。このうちSRC属性は、画像データの所在を示すために必須の属性です。

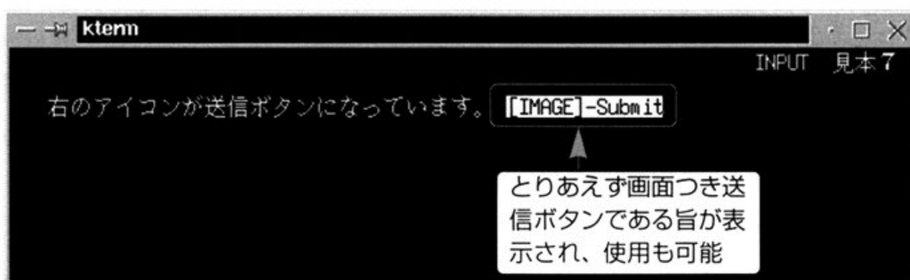
なお、HTML 3.2ではINPUT要素にALT属性が用意されていません。テキストブラウザや音声出力アプリケーションの種類によっては、送信ボタンが利用できなくなる可能性もあります。

```
<P>注文書の送信はこちら
<INPUT TYPE="image" SRC="uma.jpg" NAME="order_data"
ALIGN="middle">
</P>
```

SRC属性は必須



画像を送信ボタンにした例



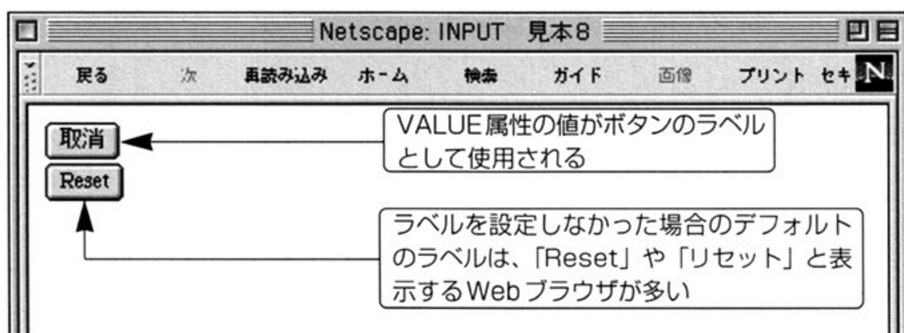
Lynxでの表示例

7. TYPE="reset"

resetは取消ボタンを表示します。取消ボタンをクリック（あるいはキー操作など）すると入力したデータがクリアされます。

詳細設定には、VALUE属性が使用できます。


```
<INPUT TYPE="reset" VALUE="取消"><BR>
<INPUT TYPE="reset">
```

TYPE属性にresetを指定した例

8. TYPE="file"

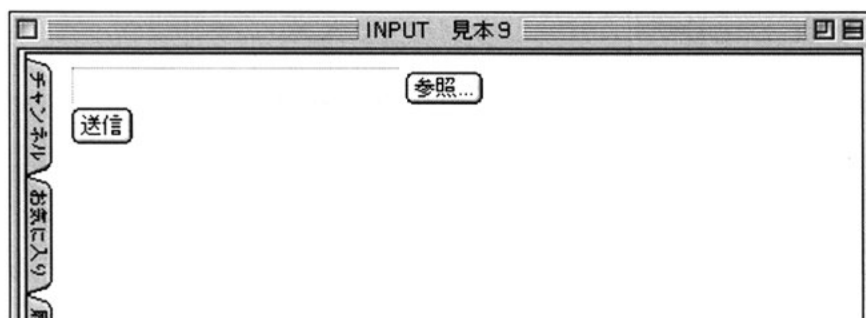
fileはファイル名の入力フィールドを表示します。

Netscape NavigatorやInternet Explorer 4.0xでは、フィールドの右側に表示される「参照...」ボタンを利用してファイルを選択します（Windows版のNetscape NavigatorやInternet Explorerでは、直接ファイル名を入力することができる）。次いで送信ボタンをクリック（あるいはキー操作など）すると、入力されたファイル名のファイルが、FORM要素のENCTYPE属性の設定に基づいて送信されます。

なお、このファイル送信機能は一般的には使用されていません。また、Lynxなど一部のWebブラウザでは対応していません。

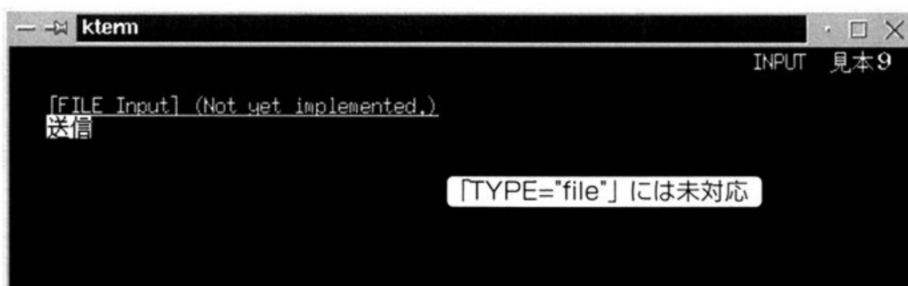
NAME属性は必須、その他詳細設定には、SIZE・MAXLENGTH属性が使用されます。

```
<INPUT TYPE="file" NAME="file_data" SIZE="30"
MAXLENGTH="30"><BR>
<INPUT TYPE="submit" VALUE="送信">
```



TYPE属性にfileを指定した例

注意 Mac OS版のNetscape Navigatorでは「参照...」ボタンが文字化けします（機能的には問題ありません）。



Lynxでの表示例

注意 HTML 3.2では、バイナリファイルの送信に対応していません。そのため、FORM要素やINPUT要素にACCEPT属性を追加し、ファイルのMIMEコンテント形式を指定する方法が提案されていました。これはHTML 4.0で正式採用されています。

```
<INPUT TYPE="file" ACCEPT="image/gif">
```

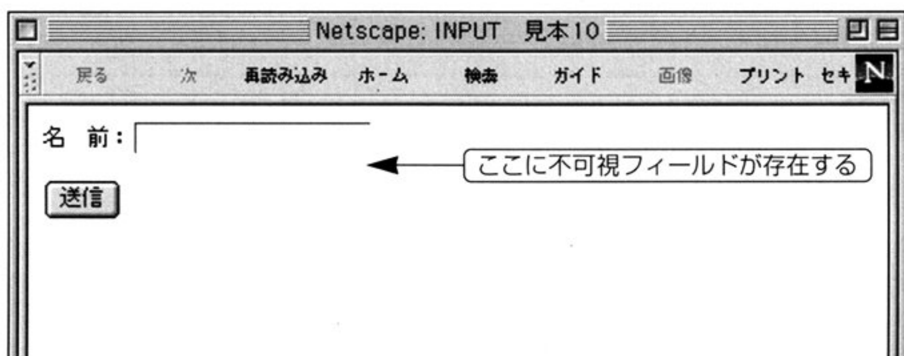
9．TYPE="hidden"

hiddenは不可視のデータフィールドを作成します。Webブラウザには何も表示されませんし、入力操作もできません。フィールドにはVALUE属性で指定されたデータが自動的に入力されます。たとえば、決まった値をCGIスクリプトに渡したい場合などに使用されます。

NAME属性は必須です。

```
名　前：<INPUT TYPE="text" NAME="namae"><BR>
<INPUT TYPE="hidden" NAME="hidden data" VALUE="
4542850590"><BR>
<INPUT TYPE="submit" VALUE="送信">
```

送付される値（固定値）



不可視フィールドを使用した例

●NAME属性

データフィールドに名前を指定します。

フォームで送信されるデータは、基本的に「入力されたデータ」だけです。そこでNAME属性を使用して、データフィールドに名前を設定し、一体何のデータであるか判別できるようにします。ほとんどのTYPE属性でNAME属性の指定が必須なのはそのためです。

```
<P> お名前・ご連絡先をお教えください。</P>
<FORM ACTION="mailto:nkiyohar@heian.go.jp" METHOD=
"POST" ENCTYPE="multipart/form-data">
  <P>
    名　前：<INPUT TYPE="text" NAME="namae"><BR>
    連絡先：<INPUT TYPE="text" NAME="contact"><BR>
           <INPUT TYPE="submit" VALUE="送信">  <INPUT
TYPE="reset" VALUE="取消">
  </P>
</FORM>
```

「名前」のフィールド名を「namae」に設定

「連絡先」のフィールド名を「contact」に設定

Netscape: INPUT　見本11

戻る　次　再読み込み　ホーム　検索　ガイド　画像　プリント　セキ

お名前・ご連絡先をお教えください。

名　前：興原敏久

連絡先：刑部省

送信　取消

このテキストをメールとして送信

```
Toshihisa O...,12:51 98.8.10 +...,Form posted from Mozilla

Subject:  Form posted from Mozilla

Date: Mon, 10 Aug 1998 12:51:07 +0900
From: Toshihisa Okihara <tokihar@heian.go.jp>
X-Mailer: Mozilla 4.03 [ja] (Macintosh; I; PPC)
To: nkiyohar@heian.go.jp
Subject: Form posted from Mozilla
Content-type: multipart/form-data; boundary=---------------------------
139541612226162
Content-Length: 265
Message-ID: <1309369041-171743839@mail.heian.go.jp>

-----------------------------139541612226162
Content-Disposition: form-data; name="namae"

興原敏久
-----------------------------139541612226162
Content-Disposition: form-data; name="contact"

刑部省
-----------------------------139541612226162--
```

NAME属性で設定したフィールド名がデータと共に送信されている

送付されたデータ

ENCTYPE="multipart/form-data"で送信されたメールの例

フォーム

●VALUE属性

データフィールドの初期値やボタンのラベル（表示名）を設定します。TYPE属性の値によって、その機能は異なります。

1. TYPE属性がtext・password・hidden（テキスト入力形式）の場合

データフィールドにデフォルトで入力される値を設定します。入力例などの表示に使用します。ただし、TYPE="hidden"の場合は、Webブラウザ上ではなにも表示されないため、ユーザは変更できません。VALUE属性で指定した値がそのまま送信されます。

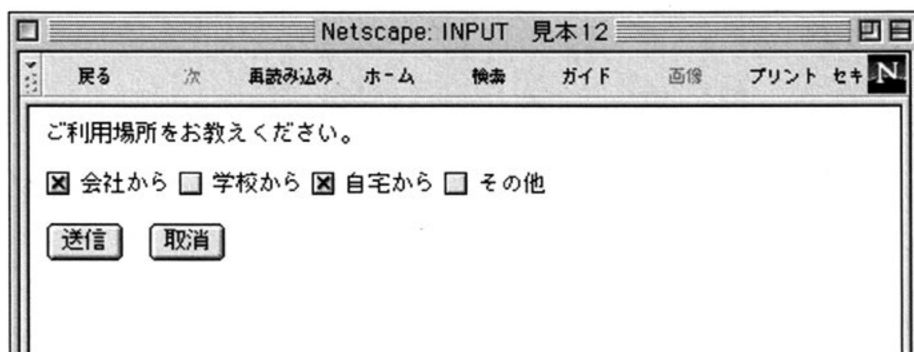
2. TYPE属性がcheckbox・radio（項目選択形式）の場合

データが送信されるとき、選択された項目はVALUE属性で設定した値に変換されます。VALUE属性を省略すると、「on」が送信されます。

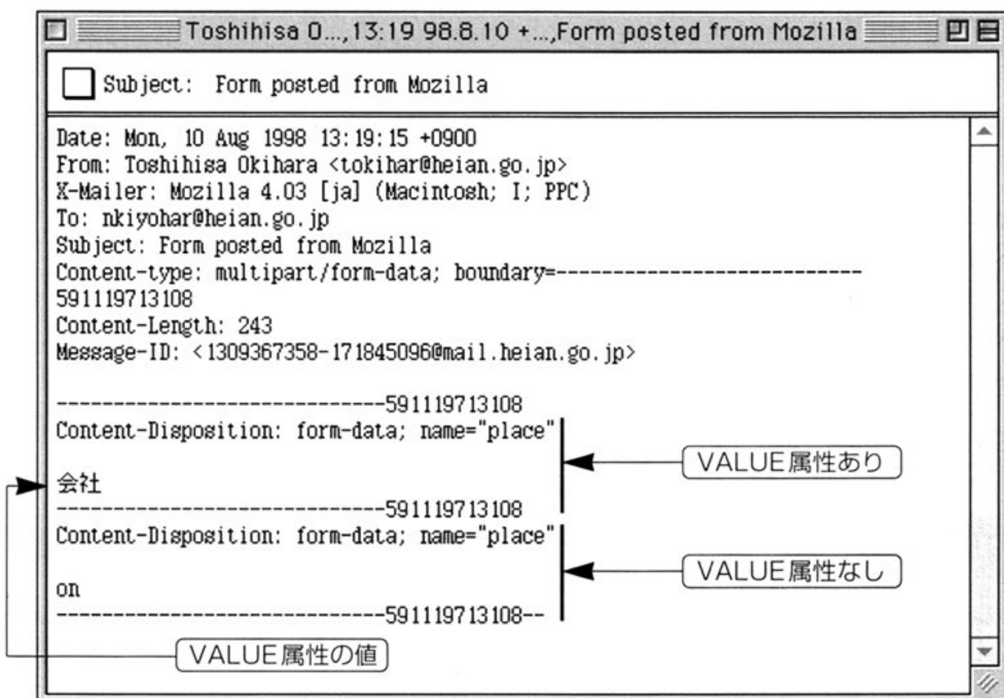
```
<P>ご利用場所をお教えください。</P>
<FORM ACTION="mailto:nkiyohar@heian.go.jp" METHOD=
"POST" ENCTYPE="multipart/form-data">
<P>
    <INPUT TYPE="checkbox" NAME="place" VALUE="会社">
会社から
    <INPUT TYPE="checkbox" NAME="place" VALUE="学校">
学校から
    <INPUT TYPE="checkbox" NAME="place">自宅から
    <INPUT TYPE="checkbox" NAME="place">その他
</P>
<P>
<INPUT TYPE="submit" VALUE="送信">  <INPUT TYPE="re-
set" VALUE="取消">
</P>
</FORM>
```

VALUE属性を設定

VALUE属性を省略



選択した内容をメールとして送信



ENCTYPE="multipart/form-data"で送信されたメールの例

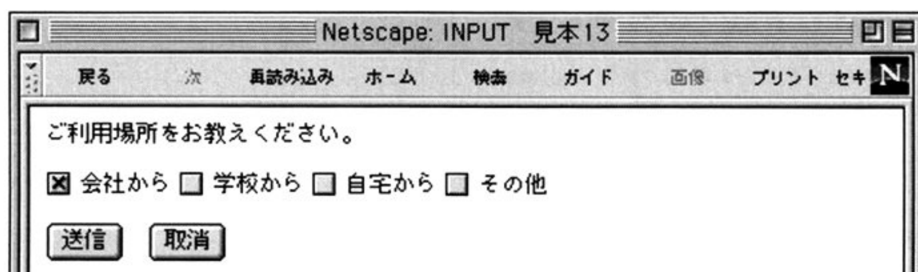
3．TYPE属性がsubmit・reset（ボタン形式）の場合

送信ボタン、取消ボタンのラベル(表示名)を設定します。

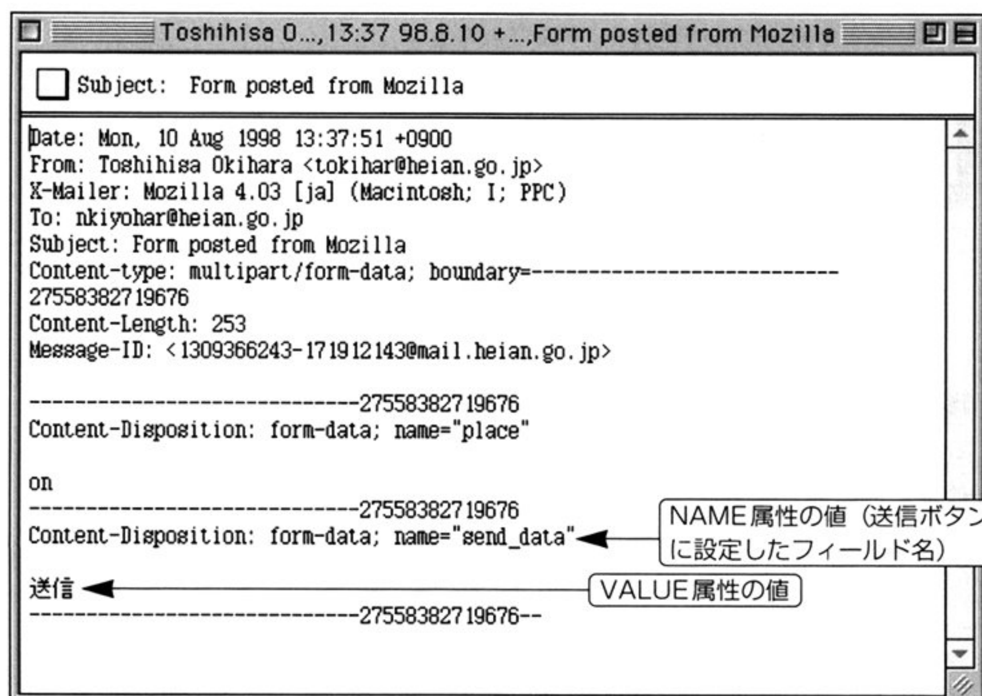
また、TYPE="submit"においてNAME属性が指定してある場合、VALUE属性の値も送信されます。

```
<P>ご利用場所をお教えください。</P>
<FORM ACTION="mailto:nkiyohar@heian.go.jp"
METHOD="POST" ENCTYPE="multipart/form-data">
<P>
  <INPUT TYPE="checkbox" NAME="place">会社から
  <INPUT TYPE="checkbox" NAME="place">学校から
  <INPUT TYPE="checkbox" NAME="place">自宅から
  <INPUT TYPE="checkbox" NAME="place">その他
</P>
<P>
  <INPUT TYPE="submit" NAME="send_data" VALUE="送信">
<INPUT TYPE="reset" VALUE="取消">
</P>
</FORM>
```

NAME属性とVALUE属性を設定



選択した内容をメールとして送信



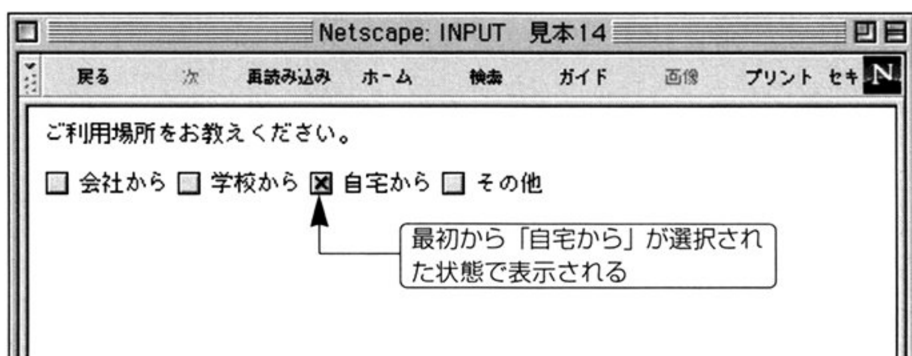
ENCTYPE=" multipart/form-data" で送信されたメールの例

●CHECKED属性

TYPE属性がcheckbox・radioで、特定の要素（項目）をデフォルトで選択状態にさせたい場合に使用します。CHECKED属性を指定する場合は属性名は省略し、属性値「checked」のみを記述します（P.8参照）。

```
<INPUT TYPE="checkbox" NAME="place">会社から
<INPUT TYPE="checkbox" NAME="place">学校から
<INPUT TYPE="checkbox" NAME="place" checked>自宅から
<INPUT TYPE="checkbox" NAME="place">その他
```

「自宅から」をあらかじめ選択された状態に設定



CHECKED属性の使用例

●SIZE属性

TYPE属性がtext・passwordの場合、データフィールドのサイズを設定します。設定は1バイト（いわゆる半角）文字数で行います。

なお、Windows版のWebブラウザでは、実際の表示サイズが設定通りにならないことがあります。

●MAXLENGTH属性

TYPE属性がtext・password・fileの場合、データフィールドに入力可能な文字数を指定します。指定は1バイト（いわゆる半角）文字数で行います。ただし、SIZE属性の値（データフィールドのサイズ）を超える値を指定しても、スクロールバーなどは表示されませんので、通常はSIZE属性の値と同じ値を指定します。

なお、TYPE属性がfileの場合、参照ボタンで選択するファイル名は

MAXLENGTH属性の制限を受けません。

●SRC属性

TYPE属性がimageの場合、画像付き送信ボタンに利用する画像のURLを指定します。

●ALIGN属性

ALIGN属性は、画像付き送信ボタンの配置位置の設定を行います。

表示結果は、IMG要素のALIGN属性(P.221参照)を参照してください。

Column　Lynxとフォーム

本文中ではレイアウトの都合もあって、テキストブラウザでのフォームの表示例をあまり載せられませんでした。その代わりといってはなんですが、P.Xの「ビジュアル索引3」をLynxで表示した例をここで紹介しておきます。

```
ktem
                              自然派雑貨屋　ベイジ　ーアンケートー
                    ご意見&アンケート
  ------------------------------------------------------------
このたびは、ベイジのWebサイトをご訪問くださいまして、誠にありがとうござ
いました。
ご意見・ご感想を頂ければ助かります。今後の参考資料とさせて頂きます。
※全項目のご記入は必要ありません。お気軽にどうぞ！

 [ご意見箱]

例）もう少し取り扱い商品の幅を広げて欲しい。_____________________
__________________________________________________________________
__________________________________________________________________
__________________________________________________________________

 [お名前]  __________________
 [ご住所]
 __________________________________________________________
 [メール]  ____________________________
 [年　齢]
 [性　別]  [あなたの性別]

 [興味を持たれた商品]
[草木染め______________]

 [ご利用の環境]  [ ] Mac OS   [X] UNIX   [ ] Windows   [ ] 各種携帯端末
   [ ] その他

 [ご利用のWebブラウザ]  (*) Netscape Navigator   ( ) Internet Explorer
   ( ) Lynx   ( ) Cyberdog   ( ) その他

送る　やり直し
  ------------------------------------------------------------
                                      自然派雑貨屋　ベイジ

(Text entry field) Enter text.  Use UP or DOWN arrows or tab to move off.
           Enter text into the field by typing on the keyboard
   Ctrl-U to delete all text in field, [Backspace] to delete a character
```

「ビジュアル索引3」をLynxで表示した例

ご覧のように、Lynxでも一般的なWebブラウザと同様、HTML 3.2のフォーム機能を利用できることがお解り頂けると思います。もちろん、グラフィカルな表示はできないので選択メニュー（SELECT要素）は直接表現できませんが、該当個所を選択すると、別途ポップアップメニューが表示され、項目（OPTION要素）を選択することができます。

なんとなく「テキストブラウザ＝低機能」と思われがちですが、Lynxはさまざまな工夫によって、他のWebブラウザに引けを取らない機能を持っています。加えて、HTMLの解釈は正確、動作も軽快、それでいてフリーウェア。UNIXユーザを中心に愛用者が多いという事実も肯けます。

SELECT

<SELECT>~</SELECT>

選択メニュー（The SELECT element）　テキストレベル要素

内容 ▶ OPTION
※1個以上

親要素 ▶ BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　PRE　DIV　CENTER　BLOCKQUOTE　FORM　CAPTION　TH　TD　DT　DD　LI
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM　STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A　APPLET　FONT

要素の概要

SELECT要素は、フォームで複数の項目から1つの項目、または複数の項目を選択するメニューに使用します。SELECT要素は、通常はFORM要素内に置かれます（テキストレベル要素なので、直接的にはさまざまな要素内に置くことができる）。

開始タグ<SELECT>と終了タグ</SELECT>で、選択メニューの範囲をマークアップします。また、終了タグ</SELECT>は省略可能です。内容としては、具体的な選択用の項目となるOPTION要素（必須）のみを置くことができます。

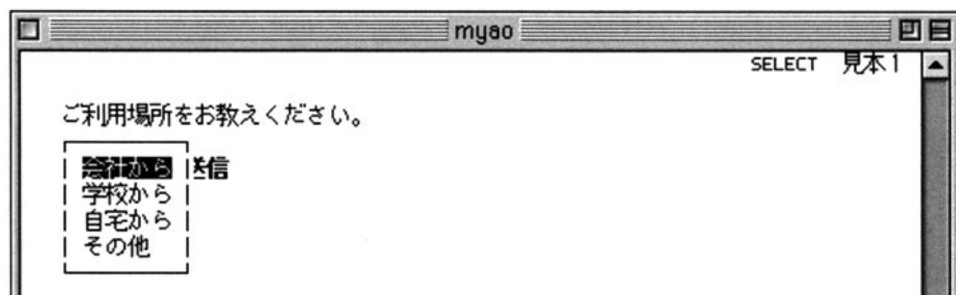
```
<P>ご利用場所をお教えください。</P>
<FORM ACTION="mailto:nkiyohar@heian.go.jp"
METHOD="POST" ENCTYPE="multipart/form-data">
<P>
 <SELECT NAME="place">
     <OPTION>会社から</OPTION>
     <OPTION>学校から</OPTION>
     <OPTION>自宅から</OPTION>
     <OPTION>その他</OPTION>
 </SELECT>
     <INPUT TYPE="submit" VALUE="送信">
</P>
</FORM>
```

他の要素の中に置くこともできる



選択メニュー表示例（ポップアップメニュー形式）

なお、Lynxなどテキストブラウザでも SELECT 要素に対応しているものもあります。



属性一覧

●NAME

値	働き	H3	N4	I4
text	データフィールド名の設定	○	○	○

属性は必須

●SIZE

値	働き	H3	N4	I4
number	選択項目の同時表示数（一覧式メニューの場合）	○	○	○

属性は省略可

●MULTIPLE（論理型）

値	働き	H3	N4	I4
multiple	複数項目の選択を許可（一覧式メニューの場合）	○	○	○

属性は省略可

変数　*text*＝一般的なテキスト　*number*＝数値

フォーム

属性解説

●NAME属性

データフィールドに名前を設定します。SELECT要素では必須の属性です。

フォームで送信されるデータは、基本的に「入力されたデータ」だけです。そこでNAME属性を使用して、データフィールドに名前を設定し、一体何のデータであるか判別できるようにします(INPUT要素のNAME属性　P.188参照)。

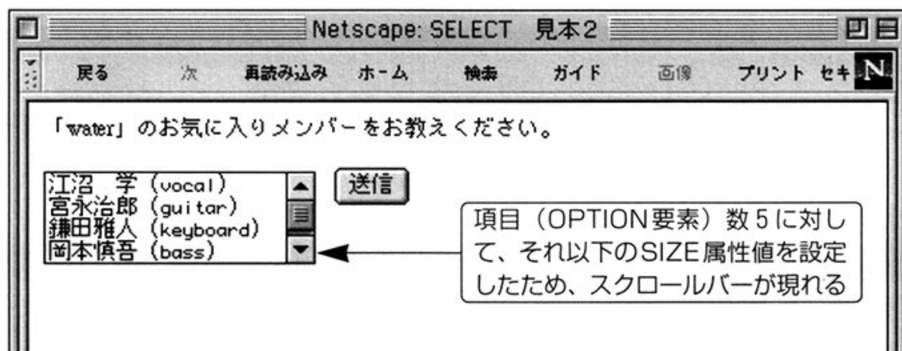
●SIZE属性

SELECT要素は、デフォルトではポップアップメニュー形式で表示されますが、SIZE属性で選択項目(OPTION要素)の同時表示数を設定すると、一覧式メニュー形式で表示されます。

属性値は表示させる行数を設定します。また、SIZE属性で設定した値より項目数が多い場合は、メニュー右端のスクロールバーが有効になります。

なお、メニューの表示幅は、最長の選択項目に自動的に合わせられます。

```
<SELECT NAME="favorite" SIZE="4">
  <OPTION>江沼　学 (vocal) </OPTION>
  <OPTION>宮永治郎 (guitar) </OPTION>
  <OPTION>鎌田雅人 (keyboard) </OPTION>
  <OPTION>岡本慎吾 (bass) </OPTION>
  <OPTION>吉田太郎 (drums) </OPTION>
</SELECT>
```

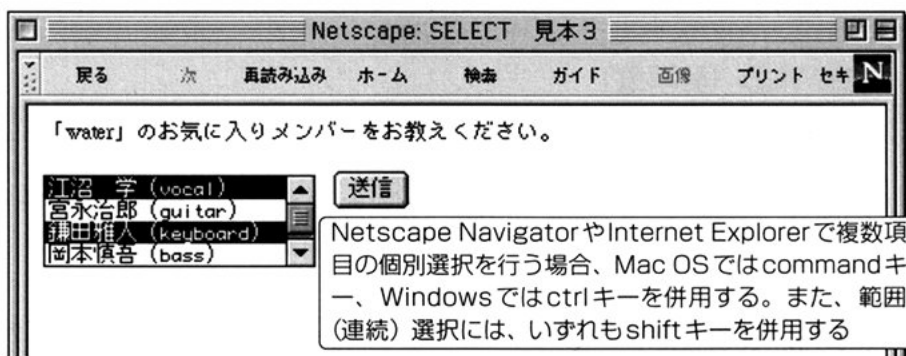


SIZE属性の使用例

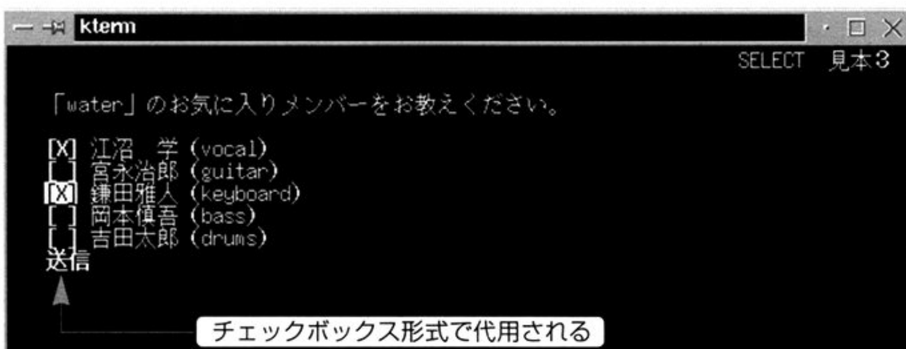
●MULTIPLE属性

SIZE属性を指定して一覧式メニューを使用した場合、MULTIPLE属性を設定することによって複数項目（OPTION要素）の選択ができるようになります。MULTIPLE属性を設定する場合は属性名は省略し、属性値「multiple」のみを記述します（P.8参照）。

```
<SELECT NAME="favorite" SIZE="4" multiple>
  <OPTION>江沼　学（vocal）</OPTION>
  <OPTION>宮永治郎（guitar）</OPTION>
  <OPTION>鎌田雅人（keyboard）</OPTION>
  <OPTION>岡本慎吾（bass）</OPTION>
  <OPTION>吉田太郎（drums）</OPTION>
</SELECT>
```



MULTIPLE属性の使用例



MULTIPLE属性の使用例

フォーム

N3 I3 N4 I4

OPTION

<OPTION>~</OPTION>

選択メニューの項目（The OPTION element）　テキストレベル要素

内容 ▶ 一般的なテキスト

親要素 ▶ SELECT

要素の概要

OPTION要素は、SELECT要素で表示される選択メニューの項目となる要素で、必ずSELECT要素内に置かれます。

開始タグ<OPTION>と終了タグ</OPTION>によって、選択メニューの項目がマークアップされます。終了タグは省略可能です。

属性一覧

●SELECTED（論理型）

値	働き	H3	N4	I4
selected	デフォルトの選択項目として設定	○	○	○

属性は省略可

●VALUE

値	働き	H3	N4	I4
text	選択項目の代わりに送信されるデータを設定	○	○	○

属性は省略可



text＝一般的なテキスト

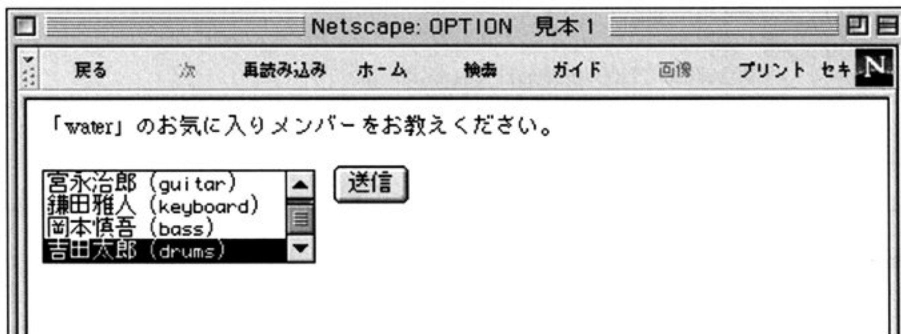
属性解説

●SELECTED属性

デフォルトで選択される項目を設定したい場合に使用されます。たとえば、あらかじめ選択する人が多いと思われる項目などで使用します。SELECTED属性を設定する場合は属性名は省略し、属性値「selected」のみを記述します(P.8参照)。

```
<SELECT NAME="place" SIZE="4">
  <OPTION>江沼　学（vocal）</OPTION>
  <OPTION>宮永治郎（guitar）</OPTION>
```

```
  <OPTION>鎌田雅人 (keyboard) </OPTION>
  <OPTION>岡本慎吾 (bass) </OPTION>
  <OPTION selected>吉田太郎 (drums) </OPTION>
</SELECT>
```



SELECTED属性の使用例

●VALUE属性

データが送信されるとき、選択された項目はVALUE属性で設定した値に変換されます。VALUE属性を省略すると、項目名(OPTION要素の内容)がそのまま送信されます。

```
<SELECT NAME="favorite" SIZE="4" multiple>
  <OPTION>江沼　学 (vocal) </OPTION>
  <OPTION>宮永治郎 (guitar) </OPTION>
  <OPTION VALUE="MK">鎌田雅人 (keyboard) </OPTION>
  <OPTION VALUE="SO">岡本慎吾 (bass) </OPTION>
  <OPTION VALUE="TY">吉田太郎 (drums) </OPTION>
</SELECT>
```

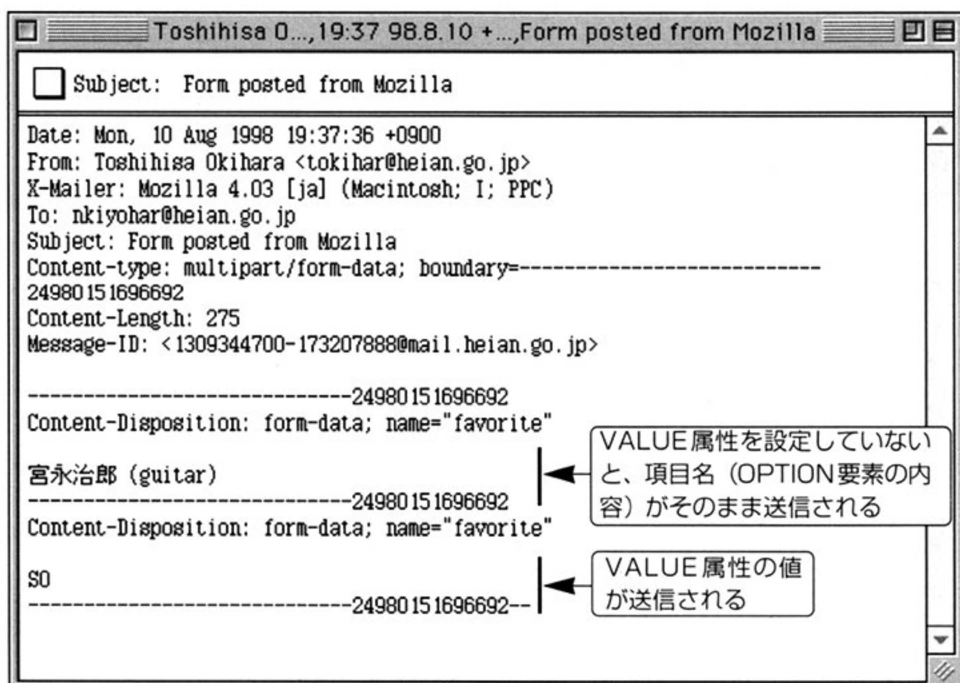
VALUE属性を設定していない項目

VALUE属性を設定した項目



VALUE属性の使用例

フォーム


Toshihisa O...,19:37 98.8.10 +...,Form posted from Mozilla

Subject: Form posted from Mozilla

```
Date: Mon, 10 Aug 1998 19:37:36 +0900
From: Toshihisa Okihara <tokihar@heian.go.jp>
X-Mailer: Mozilla 4.03 [ja] (Macintosh; I; PPC)
To: nkiyohar@heian.go.jp
Subject: Form posted from Mozilla
Content-type: multipart/form-data; boundary=---------------------------
24980151696692
Content-Length: 275
Message-ID: <1309344700-173207888@mail.heian.go.jp>

-----------------------------24980151696692
Content-Disposition: form-data; name="favorite"

宮永治郎 (guitar)
-----------------------------24980151696692
Content-Disposition: form-data; name="favorite"

SO
-----------------------------24980151696692--
```


ENCTYPE="multipart/form-data"で送信されたメールの例

TEXTAREA <TEXTAREA>~</TEXTAREA>

テキスト入力欄（The TEXTAREA element）　テキストレベル要素

内容 ▶ 一般的なテキスト

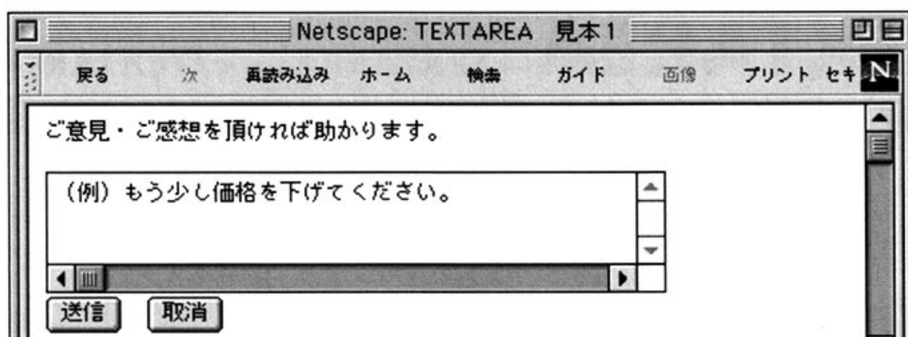
親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P PRE DIV CENTER BLOCKQUOTE FORM CAPTION TH TD DT DD LI
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

TEXTAREA要素は、フォームのテキスト入力欄の設定に使用します。必ずFORM要素内に置かれます（テキストレベル要素なので直接的にはさまざまな要素内に置くことができる）。

開始タグ<TEXTAREA>と終了タグ</TEXTAREA>でマークアップされた内容は、デフォルトで入力されるデータとなります。たとえば、記入例の表示などに使用します。

```
<P>ご意見・ご感想を頂ければ助かります。</P>
<FORM ACTION="mailto:nkiyohar@heian.go.jp"
METHOD="POST" ENCTYPE="multipart/form-data">
<P>
<TEXTAREA NAME="iken" ROWS="4" COLS="50">
（例）もう少し価格を下げてください。</TEXTAREA><BR>
<INPUT TYPE="submit" VALUE="送信"> <INPUT TYPE="re
set" VALUE="取消">
</P>
</FORM>
```



TEXTAREA要素の使用例

属性一覧

●NAME

値	働き	H3	N4	I4
text	データフィールド名の設定	○	○	○

属性は必須

●ROWS

値	働き	H3	N4	I4
number	テキスト入力欄の縦幅を行数で設定	○	○	○

属性は必須

●COLS

値	働き	H3	N4	I4
number	テキスト入力欄の横幅を行数で設定	○	○	○

属性は必須

●WRAP（初期値：off）

値	働き	H3	N4	I4
soft (vittual)	テキスト入力欄の幅に合わせて改行（画面上のみ）	−	!	!
hard (physical)	テキスト入力欄の幅に合わせて改行（データも改行）	−	!	!
off	改行なし	−	!	!

属性は省略可



number=数値　*text*=一般的なテキスト

属性解説

●NAME属性

データフィールド（テキスト入力欄）に名前を設定します。TEXTAREA要素では必須の属性です。

フォームで送信されるデータは、基本的に「入力されたデータ」だけです。そこでNAME属性を使用して、データフィールドに名前を設定し、一体何のデータであるか判別できるようにします(INPUT要素のNAME属性参照　P.188参照)。

●ROWS属性

テキスト入力欄の縦サイズを行数で設定します。TEXTAREA要素では必須の属性です。

なお、Windows版Internet Explorer 3.0xでは、設定した値と表示結果がうまく合わない場合があります。

●COLS属性

テキスト入力欄の横サイズを行数で設定します。TEXTAREA要素では必須の属性です。

なお、Windows版Internet Explorer 3.0xでは、設定した値と表示結果がうまく合わない場合があります。

●WRAP属性

テキスト入力欄内の改行方法を指定します。

soft（またはvirtual）を指定した場合、テキストが入力欄の右端に達すると画面上では改行して表示されますが、実際に送信されるデータは改行されません。

hard（またはphysical）を指定した場合、テキストが入力欄の右端に達すると画面上で改行して表示されるだけでなく、実際に送信されるデータも改行されます。

offを指定した場合、テキストが入力欄の右端に達しても改行されず、実際に送信されるデータも改行されません。

なお、WRAP属性はもともとNetscape Navigatorの独自拡張でしたが、その後Internet Explorerでも採用されました。そのせいもあって、HTML 3.0では審議の対象になりましたが、結局HTML 3.2やHTML 4.0でも採用されていません。したがって、一般のHTML文書では使用しない方がいいでしょう。

NN 3 IE 3 NN 4 IE 4

A

<A>~</A>

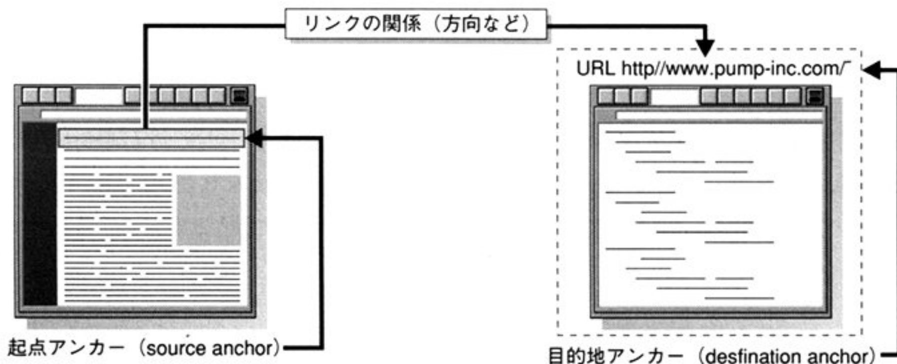
アンカー（The Anchor element）　　テキストレベル要素

内容 ▶ TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE
INPUT　SELECT　TEXTAREA　IMG　APPLET　FONT
BASEFONT　BR　MAP　SCRIPT
一般的なテキスト

親要素 ▶ BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　LI　DT　DD　PRE　DIV　CENTER
BLOCKQUOTE　FORM　CAPTION　TH　TD
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE
APPLET　FONT

要素の概要

HTML文書の最大の特徴は、その名前の由来にもなっている「ハイパーテキストリンク機能」です。このリンクとは、2つ（あるいはそれ以上の）ものを結びつける関係です。その関係を表現するためには、リンクの起点と目的地、リンクの関係（方向など）を示す3つの情報が必要です。もっとも、起点と目的地が判明していれば、関係の情報はなくてもとりあえずリンクは成立します。このリンクの起点と目的地を「アンカー（anchor）」と言います。また、特に起点側を「起点アンカー（source anchor）」、目的地側を「目的地アンカー（destination anchor）」とも言います。



リンクとアンカーの基本関係

起点アンカーは通常HTML文書の中の特定の部分に設定されます。また、目的地アンカーは目的地がWWWのリソース（HTML文書や画像など）である場合、そのURLとなりますが、目的地がHTML文書内の特定の部分である場合、目的地アンカーには別途名前がつけられます。

A要素は、HTML文書の特定の部分にこれらの「アンカー」を設定する要素です。開始タグ<A>と終了タグ</A>によって、アンカーとして設定する内容をマークアップします。また、属性によって「起点アンカー」か「目的地アンカー」、あるいはその両方にすることができます。

なお、A要素はテキストレベル要素です。したがって、アンカーを設定できる内容は、テキストレベル要素と一般的なテキストのみです。ただし、A要素を入れ子にすることはできませんので、記述の際は注意してください。

```
<P><A HREF="http://www.pump-inc.com/">PUMPのWebサイトへ
行く</A></P>
<A HREF="http://www.pump-inc.com/"><P>PUMPのWebサイトへ
行く</P></A>
```

○

× ブロックレベル要素である P要素は入れられない

属性一覧

●HREF

値	働き	H3	N4	I4
url	リンク先リソースのURL	○	○	○

属性は省略可

●NAME

値	働き	H3	N4	I4
name	目的アンカーの名前	○	○	○

属性は省略可

●REL

値	働き	H3	N4	I4
link type	順方向のリンク関係	○	－	－

属性は省略可

●REV

値	働き	H3	N4	I4
link type	逆方向のリンク関係	○	×	×

属性は省略可

●TITLE

値	働き	H3	N4	I4
text	リンク先リソースのタイトル	○	×	○

属性は省略可

●TARGET

値	働き	H3	N4	I4
top	別ウインドウを開いてリンク先を表示	–	!	!
frame-target	ターゲットにするフレーム名	–	!	!
_top	フレーム枠を解除してリンク先を表示	–	!	!

属性は省略可

変数 *url*=URL　*text*=一般的なテキスト　*link type*=文字間の関係の種類
name=一般的なテキスト（日本語使用不可　大文字小文字の区別あり）
frame-target=一般的なテキスト（先頭文字はアルファベットのみ）

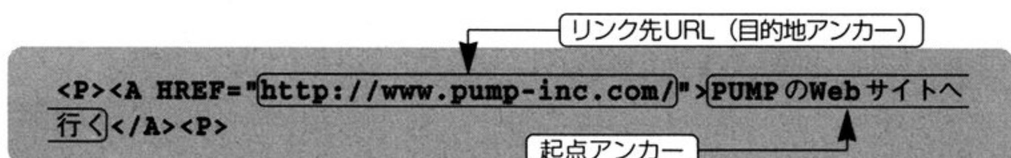
属性解説

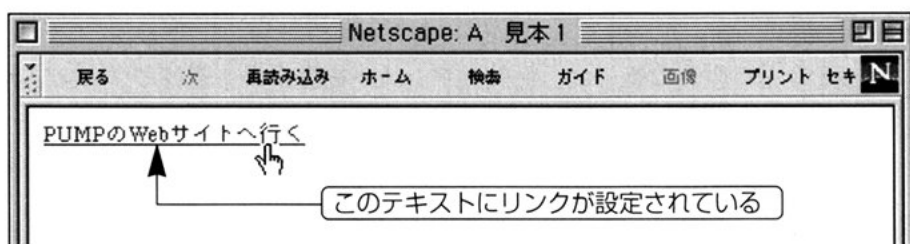
●HREF属性

HREF属性を単独で使用すると、A要素の内容は起点アンカーとして機能します。属性値にはリンク先のリソースのURLを指定します。値には、日本語は使用できません。また、大文字・小文字は区別されます。

リンク先のリソースは、HTML文書に限りません。画像や動画、PDFなどWWW上で利用可能な全てのデータを含みます。

リンクが設定されたテキスト色は、Webブラウザに設定されているものが使用されますが、HTML文書の作者がBODY要素のLINK属性で指定することもできます(P.25参照)。





A要素の使用例

また、値にメールの宛先を指定する「mailto:メールアドレス」もよく利用されます。この場合、リンクをクリックすると、Webブラウザのメールウインドウが開きます（あるいは設定されているメールソフトが起動）。メールウインドウの宛先欄には、設定されたメールアドレスが自動的に入力されます。

```
<P>ご意見・ご感想を頂ければ助かります。<P>
<ADDRESS>
<A HREF="mailto:webmaster@pump-inc.com">PUMP inc.
(webmaster@pump-inc.com) </A>
</ADDRESS>
```

値がメールアドレス



目的地アンカーがメールアドレスの場合、メーラーが起動する

リンク

●NAME属性

HTML文書内の特定の部分に、目的地アンカーを設定する属性です。NAME属性を単独で使用すると、A要素の内容は目的地アンカーとして機能します。

属性値には、目的地アンカーの「名前」を記述します。この名前にはHREF属性と同様、日本語は使用できません。大文字・小文字は区別されます。また、1つのHTML文書内に複数の目的地アンカーを設定する場合、それらは別々の名前でなければなりません。

HTML文書内に設定された目的地アンカーにリンクを設定する場合は、起点アンカーのHREF属性は、「URL」+「#」+「目的地アンカーの名前」の形式で記述します。ただし、同じ文書内に目的地アンカーがある場合はURLを省略できます。

HTML文書内の目的地アンカーは特に色が変化したりはしないので、Webブラウザで表示しても見た目では判別できません。

```
<H1>6. 職員令民部省関連条文</H1>
<P>目 次</P>
<OL>
<LI><A HREF="#6-1">民部省条</A></LI>
<LI><A HREF="#6-2">主計寮条</A></LI>
<LI><A HREF="#6-3">主税寮条</A></LI>
</OL>
<HR>
<H2><A NAME="6-1">6-1. </A>民部省条</H2>
<P>卿一人。<SMALL>諸国の戸口の名籍、賦役、孝義、優復、けん [益十蜀] 免、家人、奴婢、橋道、津済、渠池、山川、藪沢、諸国の田の事を掌る。
</SMALL>大輔一人。少輔一人。大丞一人。少丞二人。大録一人。少録三人。
史生十人。省掌二人。使部六十人。直丁四人。</P>
<H2><A NAME="6-2">6-2. </A>主計寮条</H2>
<P>頭一人。<SMALL>調及び雑物を計り納れ、国用を支度し、用度を勘勾するを掌る。</SMALL>助一人。大充一人。少充二人。大属一人。少属一人。
算師二人。<SMALL>調庸及び用度を勘計するを掌る。</SMALL>史生六人。
使部廿人。直丁二人。</P>
<H2><A NAME="6-3">6-3. </A>主税寮条</H2>
<P>頭一人。<SMALL>倉廩、出納、諸国の田租、舂米、碾磑の事を掌る。
</SMALL>助一人。大充一人。少充二人。大属一人。少属一人。算師二人。
<SMALL>租税を勘計するを掌る。</SMALL>史生四人。使部廿人。直丁二
人。</P>
```

目的地アンカーが同じ文書内にあるのでURL省略

目的地アンカーの名前

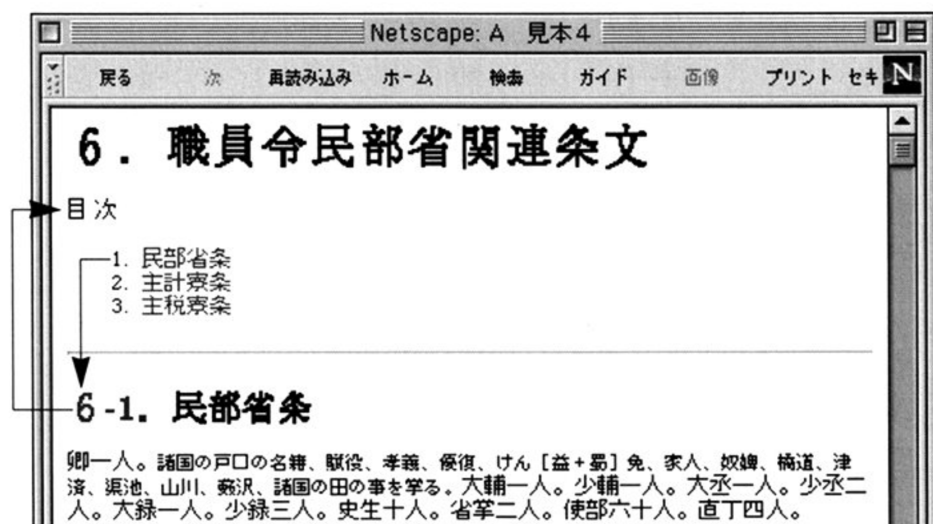
目的地アンカー



ページ内リンクの使用例

また、NAME属性とHREF属性は同時に設定することもできます。その場合のA要素は、目的地アンカーであると同時に起点アンカーとしても機能します。

```
<H1>6．職員令民部省関連条文</H1>
<P><A NAME="mokuji">目 次</A></P>
<OL>
<LI><A HREF="#6-1">民部省条</A></LI>
<LI><A HREF="#6-2">主計寮条</A></LI>
<LI><A HREF="#6-3">主税寮条</A></LI>
</OL>
<HR>
<H2><A NAME="6-1" HREF="#mokuji">6-1.</A>民部省条
</H2>
(以下略)
```



ページ内リンクの使用例2

さらに、別のHTML文書内に設定された目的地アンカーにリンクを設定する場合は、HREF属性の値にHTML文書自体の目的地アンカーに続けて「#」記号と目的地アンカーの「名前」を記述します。

```
<H1><A HREF="shikiin_index.html#taminotsukasa"> 6.
</A>職員令民部省関連条文</H1>
(以下略)
```

なお、A要素は属性が異なっていても入れ子にすることはできません。

```
<H2>
<A NAME="6-1"> 6-1. <A HREF="yougo.html#taminotsukasa">
民部省</A>条</A>
</H2>
```

A要素は入れ子にできない

●REL属性

元の文書（起点アンカーを設定した文書）とHREF属性で設定したリンク先文書との関係を、元の文書を基準に定義する属性です。

たとえば、リンク先文書を「次の文書（REL="next"）」と定義し、Webブラウザにバックグラウンドでリンク先文書にアクセスさせ（いわゆるデータの先読み）、表示時間の短縮を図るような使い方が想定できます。ただし、現在のところこの属性をサポートするWebブラウザはないようです。

●REV属性

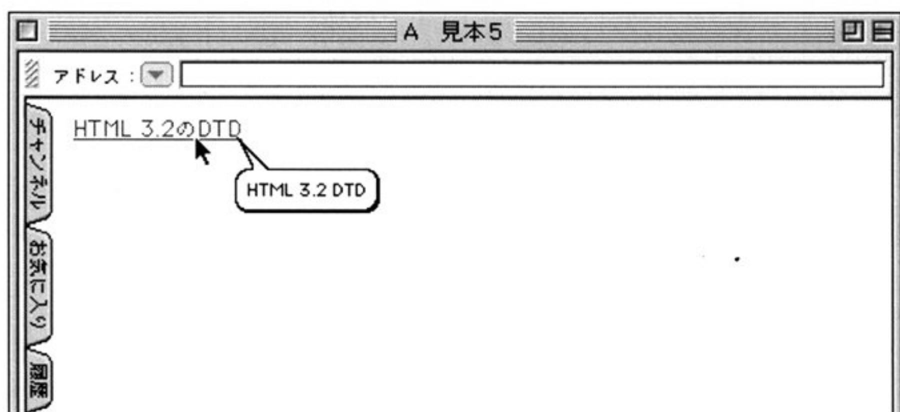
元の文書(起点アンカーを設定した文書)と、HREF属性で設定したリンク先文書との関係を、REL属性とは反対にリンク先文書を基準に定義する属性です。REL属性と同様、現在のところこの属性をサポートするWebブラウザはないようです。

●TITLE属性

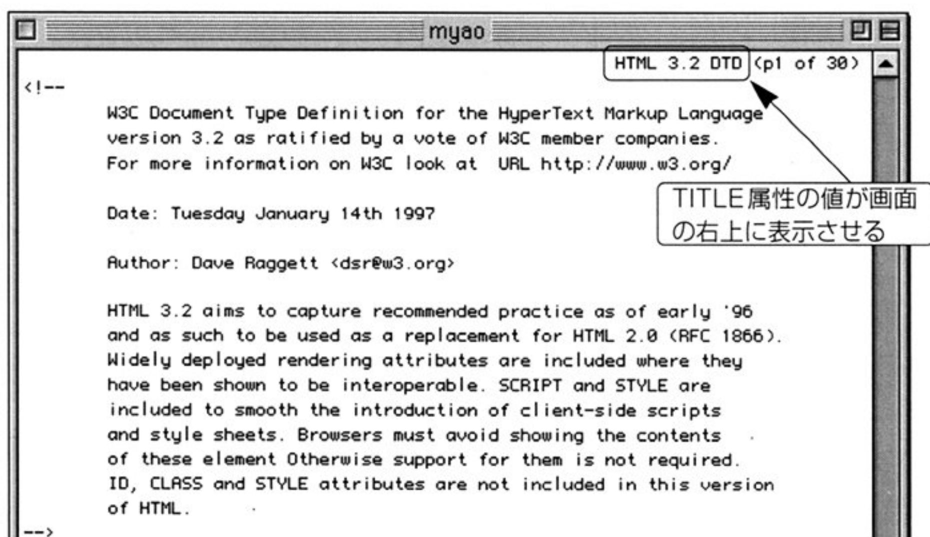
HREF属性で設定したリンク先リソースのタイトル(情報)を設定しますが、実際の表現方法に関しては、Webブラウザに任されています。

たとえば、Internet Explorer 4.0xでは、アンカー部分へマウスポインタを移動させた場合に、吹き出しでTITLE属性の値を表示させます(将来的にはこの情報を音声出力するWebブラウザも出てくるでしょう)。また、Lynxのように HTML文書以外のリソース(生のテキスト文書など)へリンクした場合、この属性の値をリソースのタイトルとして使用するWebブラウザもあります。あるいは「mailto:メールアドレス」を設定した場合に、自動的にメールのタイトルに入力させるように使用されることがあります。

```
<P><A HREF="HTML32_DTD.txt" TITLE="HTML 3.2 DTD">HTML
3.2のDTD</A></P>
```



Internet Explorer 4.0xでのTITLE属性の値の表示例



Lynxでリンク先のリソースが生のテキスト文書の場合の表示例

●TARGET属性

フレーム機能を使用する場合のウインドウの表示方法などを定義します（P.261参照）。

IMG

<IMG>（終了タグなし）

画像（inline IMaGes）　テキストレベル要素

内容 空

親要素 BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　DIV　CENTER　BLOCKQUOTE　FORM　CAPTION
TH　TD　DT　DD　LI
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
APPLET　FONT

要素の概要

IMG要素は、HTML文書内に画像を表示する場合に使用します。といっても、HTML文書自体はテキストファイルなので直接画像を含めることはできません。そこで、IMG要素には本物の画像データのURLを記すためのSRC属性が用意されており、HTML文書を表示するWebブラウザは、その値に基づいて画像データを入手し、ウインドウに表示します(あるいは、代替テキストを表示したり、音声出力したりします)。

また、IMG要素はテキストレベル要素なので、ブロックレベル要素やテキストレベル要素、あるいは一般的なテキストの中に置くことができます。

なお、IMG要素は開始タグ<IMG>のみでマークアップされる、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

属性一覧

●SRC

値	働き	H3	N4	I4
url	表示する画像のURL	○	○	○

属性は必須

●ALT

値	働き	H3	N4	I4
text	代替テキスト	○	○	○

属性は省略可（できるだけ有効に使用してください）

●ALIGN（初期値：bottom）

値	働き	H3	N4	I4
top	テキスト行上端に画像の上端を揃えて配置	○	○	○
middle	テキスト行下端に画像の中央を揃えて配置	○	○	○
bottom	テキスト行下端に画像の下端を揃えて配置	○	○	○
left	テキスト行の左端に画像を配置し、回り込みを設定	○	○	○
right	テキスト行の右端に画像を配置し、回り込みを設定	○	○	○
texttop	同一行内のテキストの上端に揃える	−	!	−
absmiddle	同一行内の最大の画像の中央に揃える	−	!	−
baseline	テキストを下付き表示	−	!	−
absbottom	同一行内の最大の画像の下端に揃える	−	!	−

属性は省略可

●WIDTH

値	働き	H3	N4	I4
pixels	画像の横幅	○	○	○

属性は省略可

●HEIGHT

値	働き	H3	N4	I4
pixels	画像の高さ	○	○	○

属性は省略可

●BORDER

値	働き	H3	N4	I4
pixels	リンク設定時の枠線の幅	○	○	○

属性は省略可

●HSPACE

値	働き	H3	N4	I4
pixels	画像の左右の余白	○	○	○

属性は省略可

●VSPACE

値	働き	H3	N4	I4
pixels	画像の上下の余白	○	○	○

属性は省略可

●USEMAP

値	働き	H3	N4	I4
url#map	クライアントサイド・イメージマップの設定	○	○	○

属性は省略可

●ISMAP（論理型）

値	働き	H3	N4	I4
ismap	サーバサイド・イメージマップの使用宣言	○	○	○

属性は省略可

●DYNSRC

値	働き	H3	N4	I4
url	表示する動画のURL	−	−	!

属性は省略可

●START

値	働き	H3	N4	I4
fileopen	ファイルが開かれると同時に再生開始	−	−	!
mouseover	マウスが移動すると同時に再生開始	−	−	!

属性は省略可

●CONTROLS（論理型）

値	働き	H3	N4	I4
controls	動画のコントロールバー表示	−	−	!

属性は省略可

●LOOP

値	働き	H3	N4	I4
number	動画再生の繰り返し回数	−	−	!
infinite	動画再生を無限に繰り返す（=-1）	−	−	!

属性は省略可

●LOOPDELAY

値	働き	H3	N4	I4
number	動画再生の繰り返し間隔	−	−	!

属性は省略可

変数 *url*＝URL　*text*＝一般的なテキスト　*pixels*＝pixel数
map＝イメージマップ名（一般的なテキスト）　*number*＝数値

属性解説

●SRC属性

表示させる画像のURLを設定します。IMG要素必須の属性です。

```
<IMG SRC="kiseki.jpg">
```

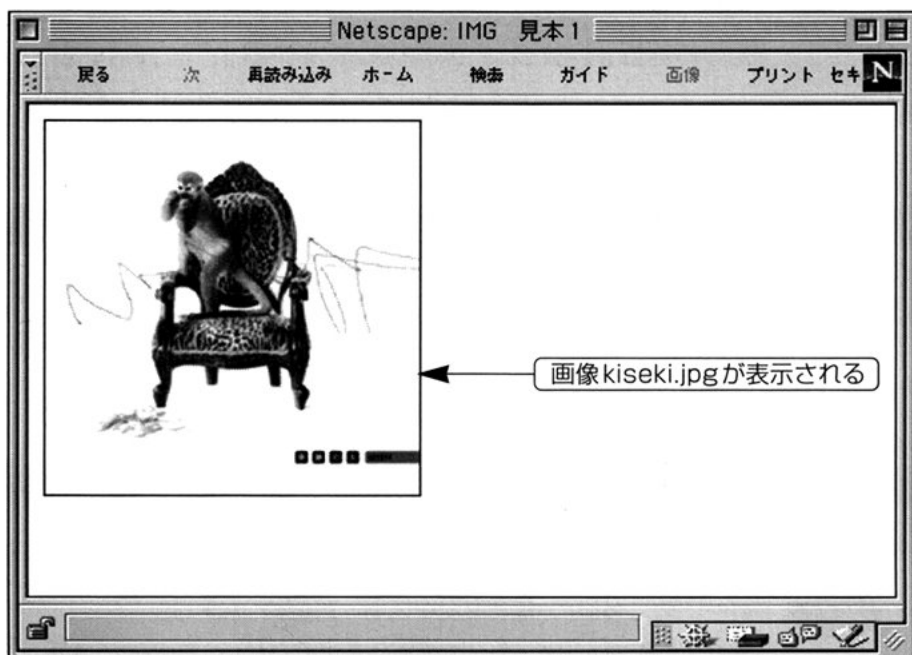
●ALT属性

主に画像を直接表示することができないWebブラウザや音声出力アプリケーションのために、代替テキストを設定する属性です。また、画像表示ができるWebブラウザでも、吹き出しなどで表示可能なものもありますし、あるいは画像表示機能をオフにしている場合にはALT属性の値が表示されます。

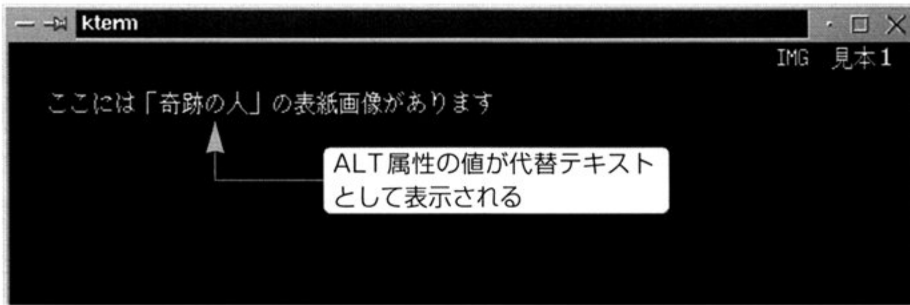
HTML 3.2では必須の属性ではありませんが、さまざまな利用方法がありますので、できるだけ有効に使用してください。

```
<IMG SRC="kiseki.jpg" ALT="ここには「奇跡の人」の表紙画像があります">
```

※SRC・ALT属性以外の属性は省略



IMG要素の使用例

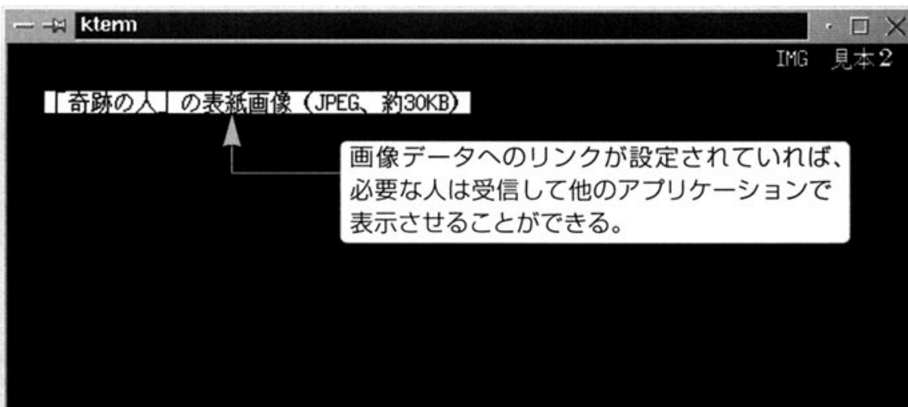


Lynxでの表示例

また、画像を表示できないWebブラウザでも、画像のデータを受信して別のアプリケーションで表示することが可能な場合もあります。そのため、重要な画像や是非見てもらいたい画像は、ALT属性の使用と同時に、画像データへのリンクを設定しておくといいでしょう。

```
<A HREF="kiseki.jpg">
<IMG SRC="kiseki.jpg" ALT="「奇跡の人」の表紙画像（JPEG、約
30KB）">
</A>
```

※SRC・ALT属性以外の属性は省略



リンクを記述してあるので別途画像の取得が可能

ただし、HTML文書の内容によっては、無理にALT属性の値を表示させない方がいい場合もあります。たとえば、単なる飾りのためのイラストなどはALT属性でくどくど説明されても、かえって困ります。そのような場合は、ALT属性を省略するのではなく、画像が存在すること自体がわからないように空の値を設定します。

```
<IMG SRC="logo2.gif" ALT="">
```

※SRC・ALT属性以外の属性は省略



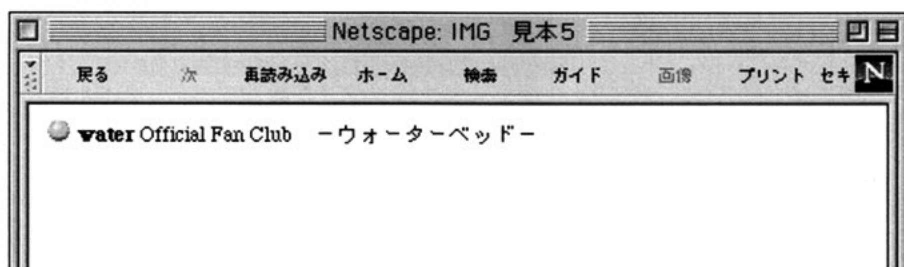
ALT属性に空の値を設定した例

あるいは、行頭などに飾りとして、またはリンク用のボタンとして使用している画像なども、無理に説明を入れず、適当な代替記号を設定するといいでしょう。

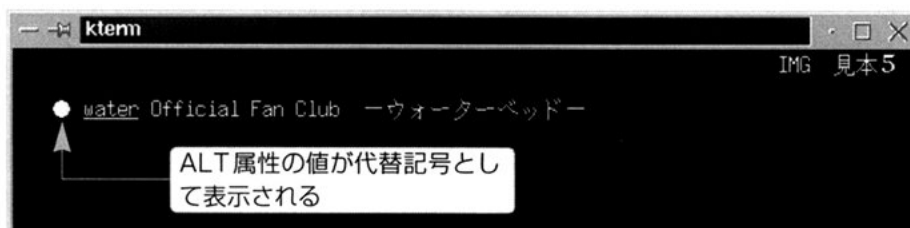
画像と似た代替記号

```
<IMG SRC="button.gif" ALT="●"><B>water</B> Official
Fan Club　ーウォーターベッドー
```

※SRC・ALT属性以外の属性は省略。



Netscape Navigatorでの表示例



Lynxでの表示例

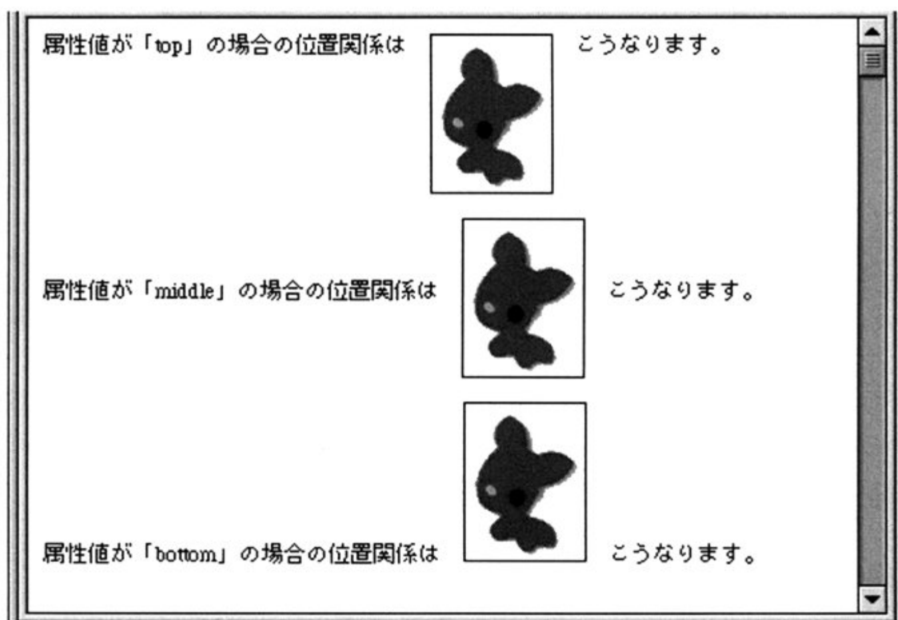
●ALIGN属性

画像と画像が含まれるテキスト行との位置関係を設定します。ただし、Webブラウザや表示フォントによっては、表現結果が微妙に異なる場合があります。

属性値top、middle、bottomは、テキスト行に対して画像を揃える位置を設定します。また、ALIGN属性を省略すると、初期値としてbottomが使用されます。

```
<P>属性値が「top」の場合の位置関係は <IMG SRC="uma2.jpg"
ALIGN="top"> こうなります。</P>
<P>属性値が「middle」の場合の位置関係は <IMG SRC="uma2.jpg"
ALIGN="middle"> こうなります。</P>
<P>属性値が「bottom」の場合の位置関係は <IMG SRC="uma2.jpg"
ALIGN="bottom"> こうなります。</P>
```

※SRC・ALIGN属性以外の属性は省略。



ALIGN属性の指定例

属性値leftとrightは、テキスト行の中での画像の配置位置（左右）を設定します。また、同時に画像に対するテキストの回り込みが設定されます。このテキストの回り込みは、画像を含む要素内だけでなく、続く別の要素にも有効となります。

```
<P><IMG SRC="kiseki.jpg" ALIGN="right"><B>「奇跡の人」
water (TOCT-10271) </B></P>
<P>コイツは、手がつけられない!!<BR>
全曲シングルかと思わせるクオリティ。<BR>
他のバンドとはひとまわり違うスケール。<BR>
R&amp;B をベースにした実力派ブリットポップバンド water の1st アルバ
ム。</P>
```

※SRC・ALIGN属性以外の属性は省略



属性値"right"の指定例

なお、属性値texttop、absmiddle、baseline、absbottomは、同じ行内に複数の画像がある場合の整列を設定します。ただし、Netscape Navigatorの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●WIDTH属性

HEIGHT属性と対になって、表示する画像の横幅をpixel数で設定します。また、原寸より大きい、あるいは小さい値を設定することもできます。その場合、画像は設定された値に合わせて拡大・縮小表示されます。ただ、Webブラウザに余計な処理をさせることになりますので、原寸以外の設定はあまりおすすめできません。

属性を省略すると、画像は原寸(画像データが持っているpixel数)で表示されます。必須の属性ではないため、一見不要な属性にも思われがちですが、この属性があるとWebブラウザは画像データを受信する前に、画像の表示サイズを把握できる

ため、先に表示領域を確保し、他のテキストなどの表示を開始することができます。反対にこの属性がないと、Webブラウザは画像データをある程度受信するまで確保すべき表示領域が把握できないため、それだけ表示が遅くなります。また、画像表示機能をオフにしている場合、画像の表示領域が確保できないため、表示が大きく乱れる可能性もあります。

●HEIGHT属性

WIDTH属性と対になって、表示する画像の高さをpixel数で設定します。

```
①→ <IMG SRC="jdm1.gif" WIDTH="240" HEIGHT="170">
②→ <IMG SRC="jdm1.gif">
```

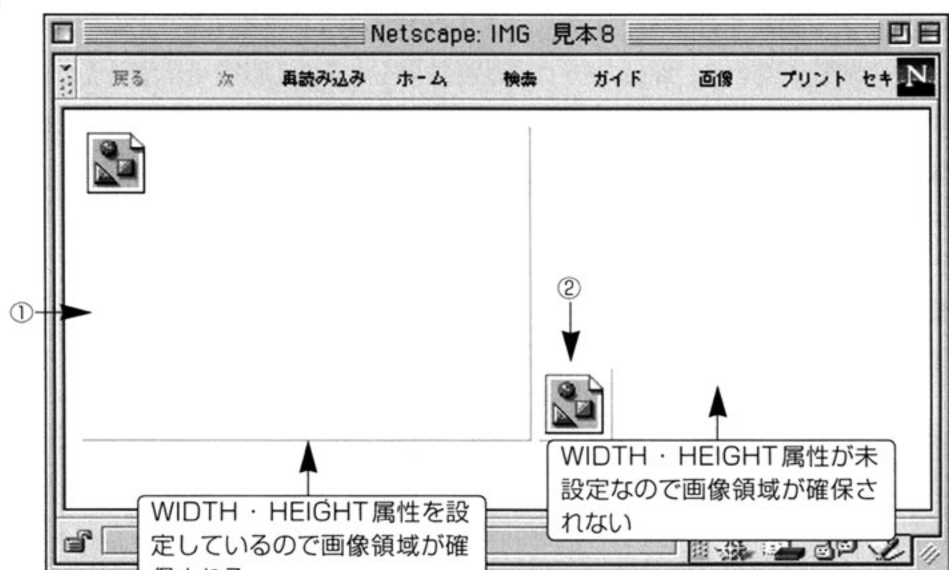
WIDTH・HEIGHT属性を設定

WIDTH・HEIGHT属性を未設定

※SRC・WIDTH・HEIGHT属性以外の属性は省略



WIDTH・HEIGHT属性の指定例



画像表示機能をオフにした場合の表示例

●BORDER属性

IMG要素をリンク(起点アンカー)の一部、または全部に設定した場合、画像の周囲に色付きの枠線が表示されます。BORDER属性はこの枠線の太さをpixel数で設定する属性です。

```
<A HREF="http://www.pump-inc.com/waterbed/index.html">
<IMG SRC="logo_new.gif" BORDER="2"></A>属性値が「2」の場
合の枠線。
<A HREF="http://www.pump-inc.com/waterbed/index.html">
<IMG SRC="logo_new.gif" BORDER="4"></A>属性値が「4」の場
合の枠線。
```

※SRC・BORDER属性以外の属性は省略



BORDER属性の使用例

また、BORDER="0"を設定すると枠線は非表示になりますが、一般的にはマウスポインタの形状などが変化するので、リンクが設定されていることは把握できます。しかし、テキストブラウザや音声出力アプリケーションのことも考慮して、やはりALT属性にその旨を記しておくことを強くおすすめします。

```
<A HREF="http://www.pump-inc.com/waterbed/index.html">
<IMG SRC="logo_new.gif" BORDER="0" ALT="「waterbed」へ
のリンクボタン">
</A>
```

※SRC・BORDER・ALT属性以外の属性は省略



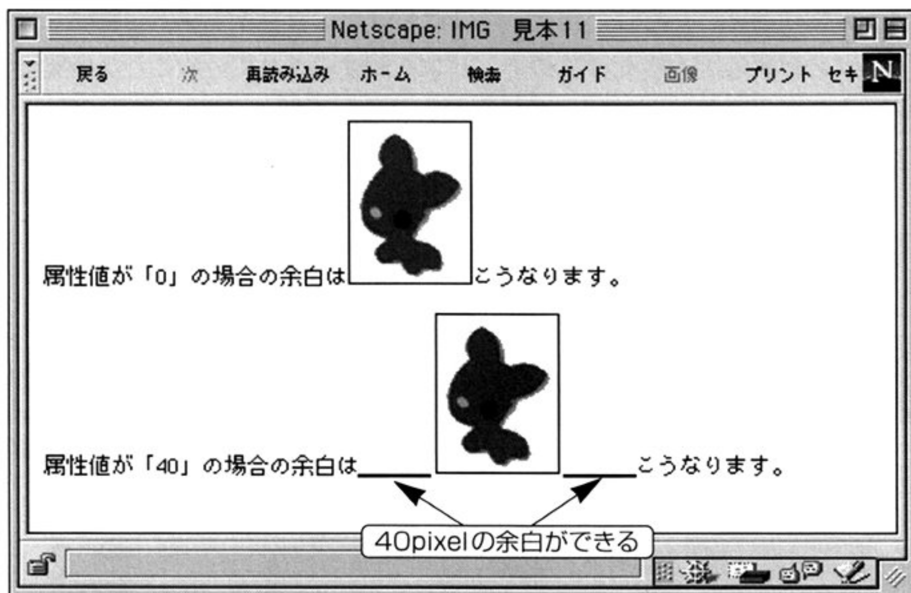
Windows版Internet Explorerでの表示例

●HSPACE属性

画像の左右に余白をpixel数で設定します。属性を省略すると、0以外の最小値が設定されます(具体的な値はWebブラウザによって異なる)。

```
<P>属性値が「0」の場合の余白は<IMG SRC="uma2.jpg" HSPACE=
"0">こうなります。</P>
<P>属性値が「40」の場合の余白は<IMG SRC="uma2.jpg" HSPACE=
"40">こうなります。</P>
```

※SRC・HSPACE属性以外の属性は省略



Netscape Navigatorでの表示例

●VSPACE属性

画像の上下に余白をpixel数で設定します。属性を省略すると、0以外の最小値が設定されます(具体的な値はWebブラウザによって異なる)。

●USEMAP属性

クライアントサイド・イメージマップ(P.229参照)を使用する場合、イメージマップの名前を指定します。

●ISMAP属性

サーバサイド・イメージマップを使用するときに使用されます。属性名は省略し属性値「ismap」のみを記述します。

現在ではほとんどのWebブラウザがクライアントサイド・イメージマップに対応していること、あるいはテキストブラウザや音声出力アプリケーションで利用しがたいことなどから、使用する機会はあまりないでしょう。

●DYNSRC属性

AVI形式の動画やVRMLを表示する際に用いられます。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●START属性

DYNSRC属性で表示される動画の再生開始方法を指定します。

fileopenはファイルのオープンと同時に再生が開始されます。mouseoverはマウスカーソルが動画上に移動したときに再生が開始されます。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●CONTROLS属性

DYNSRC属性で表示される動画にコントロールバーを付加します。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●LOOP属性

DYNSRC属性で表示される動画の再生を繰り返す回数を設定します。また、infiniteか-1を設定すると、無限に繰り返します。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

●LOOPDELAY属性

LOOP属性を設定した場合の再生を繰り返す間隔を設定します。

Internet Explorerの独自拡張なので、一般のHTML文書では使用しない方がいいでしょう。

NN 3 IE 3 NN 4 IE 4

MAP

<MAP>~</MAP>

クライアントサイド・イメージマップ（Client-side image MAPs）　テキストレベル要素

内容 AREA
※１個以上

親要素 BODY
H1　H2　H3　H4　H5　H6
ADDRESS
P　DIV　CENTER　BLOCKQUOTE　FORM　CAPTION
TH　TD　DT　DD　LI
TT　I　B　U　STRIKE　BIG　SMALL　SUB　SUP　EM
STRONG　DFN　CODE　SAMP　KBD　VAR　CITE　A
APPLET　FONT

要素の概要

MAP要素は、クライアントサイド・イメージマップ機能を実現するための要素です。

かつて「イメージマップ」、つまり１枚の画像内に複数のリンクを設定するためには、サーバサイド・イメージマップを使用していました。しかし、サーバ側でCGIスクリプトを動作させなければならないことや、テキストブラウザや音声出力アプリケーションでの利用が不可能という問題がありました。

一方、クライアントサイド・イメージマップはその名の通り、Webブラウザ側だけでイメージマップ機能を実現するものです。現在ではほとんどのWebブラウザやアプリケーションが対応していますので、比較的容易にイメージマップを使用できるようになりました。

開始タグ<MAP>と終了タグ</MAP>によって、具体的なイメージマップの範囲をマークアップします。また、その内容としてはAREA要素のみを置くことができます。

なお、MAP要素はテキストレベル要素なので、ブロックレベル要素やテキストレベル要素、あるいは一般的なテキストの中に置くことができます。

属性一覧

●NAME

値	働き	H3	N4	I4
map	イメージマップの名前	○	○	○

属性は必須

変数 *map*＝イメージマップ名（一般的なテキスト）

属性解説

●NAME属性

IMG要素のUSEMAP属性で設定したクライアントサイド・イメージマップの名前を設定し、IMG要素とMAP要素を関連づけます（実際の表示に使用する画像データのURLは、IMG要素のSRC属性で設定）。MAP要素必須の属性です。

```
<IMG SRC="cd_index.gif" USEMAP="#cd_index">
<MAP NAME="cd_index">
<AREA SHAPE="rect" COORDS="20,160,220,210" HREF=
"single.html">
<AREA SHAPE="rect" COORDS="330,240,510,340"
HREF="cd.html">
</MAP>
```

※SRC・USEMAP・NAME属性以外の属性は省略

NN 3 IE 3 NN 4 IE 4

AREA

<AREA>（終了タグなし）

クライアントサイド・イメージマップ（Client-side image map AREA）　テキストレベル要素

内容 ▶ 空

親要素 ▶ MAP

要素の概要

AREA要素は、クライアントサイド・イメージマップを使用する際に必要な座標系データやリンク、代替テキストの設定のために使用する要素です。また、必ずMAP要素の中に置かれます。

なお、AREA要素は開始タグ<AREA>のみでマークアップされる、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

属性一覧

●SHAPE（初期値：rect）

値	働き	H3	N4	I4
rect	起点アンカーとなる領域を四角形に設定	○	○	○
circle	起点アンカーとなる領域を円形に設定	○	○	○
poly	起点アンカーとなる領域を多角形に設定	○	○	○

属性は省略可

●COORDS

値	働き	H3	N4	I4
length	起点アンカーとなる領域の座標データ	○	○	○

属性は省略可

●HREF

値	働き	H3	N4	I4
url	リンク先リソースのURL	○	○	○

属性は省略可

●NOHREF（論理型）

値	働き	H3	N4	I4
nohref	リンクを無効に設定する	○	○	○

属性は省略可

●ALT

値	働き	H3	N4	I4
text	代替テキスト	○	○	○

属性は必須

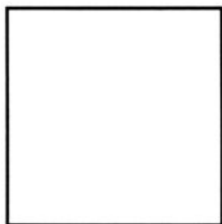


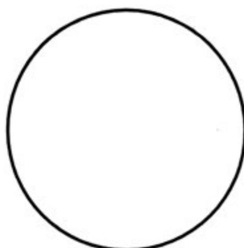
length＝pixel数、または%　*url*＝URL　*text*＝一般的なテキスト

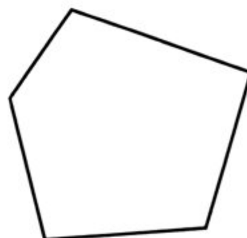
属性解説

●SHAPE属性

ハイパーテキストリンクの起点アンカーとなる領域の形状を設定します。属性値rectは四角形、circleは円形、polyは多角形です。また、属性を省略した場合はrectが使用されます。


SHAPE="rect"


SHAPE="circle"


SHAPE="poly"

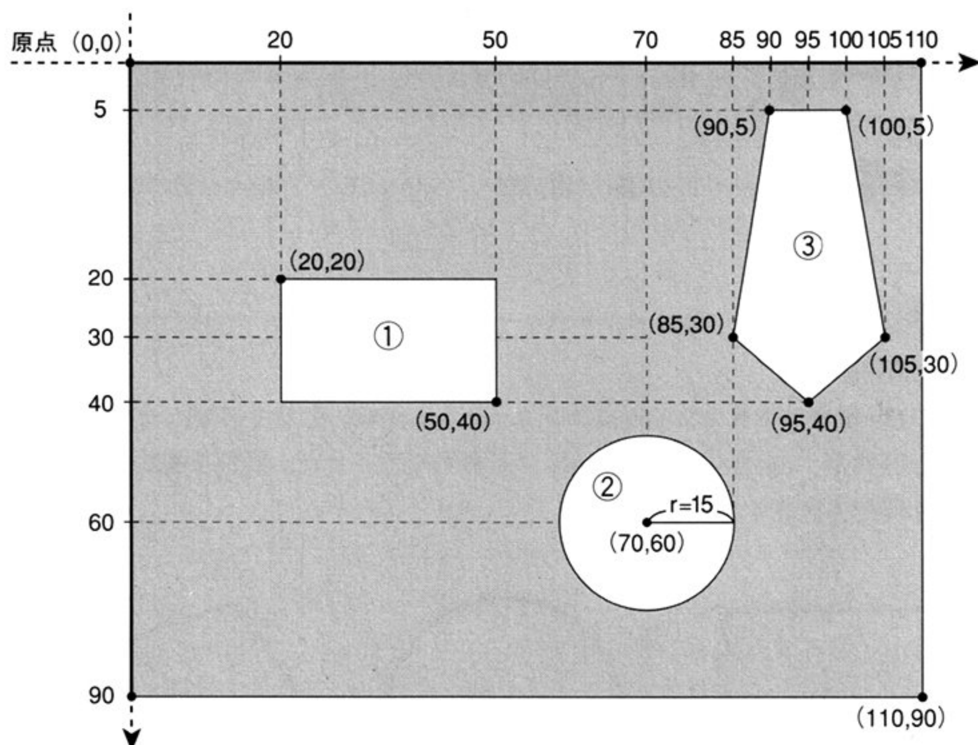
●COORDS属性

SHAPE属性で指定された形状に基づいて、実際の起点アンカー領域をpixel単位、または画像の横幅・高さに対する割合（%単位）の座標で設定します。また、座標の原点はイメージマップに使用される画像の左上になります。

SHAPE="rect"の場合は四角形の左上と右下の2ヶ所、SHAPE="cilcle"の場合は円の中心点と半径、SHAPE="poly"の場合は多角形の左上から反時計回りに各頂点の座標を記述します。

```
<AREA SHAPE="rect" COORDS="20,20,50,40">
<AREA SHAPE="circle" COORDS="70,60,15">
<AREA SHAPE="poly" COORDS="90,5,85,30,95,40,
105,30,100,5">
```

※SHAPE・COORDS属性以外の属性は省略



なお、COORDS属性で設定された領域が重なり合った場合は、先に設定された領域が有効になります。たとえば、上記の図で行の最後に画像全体をCOORDS属性で設定すると、残りの場所に領域が指定できます。

```
<AREA SHAPE="rect" COORDS="20,20,50,40">
<AREA SHAPE="circle" COORDS="70,60,15">
<AREA SHAPE="poly" COORDS="90,5,85,30,95,40,105,30,
100,5">
<AREA SHAPE="rect" COORDS="0,0,110,90">
```

※SHAPE・COORDS属性以外の属性は省略

座標による領域設定は結構面倒なのですが、簡単に領域設定できるユーティリティがオンライン上で入手できます。

◇ImageMapper（Mac OS用）

作者：Stuart Snaddon氏（日本語化：Bridge 1 Software）

価格：1,000円

入手先　：http://www.bridge1.com/
　　　　　http://www.vector.co.jp/　他

◇**Mapedit（Windows用）**

作者：Boutell.Com, Inc.

価格：$25

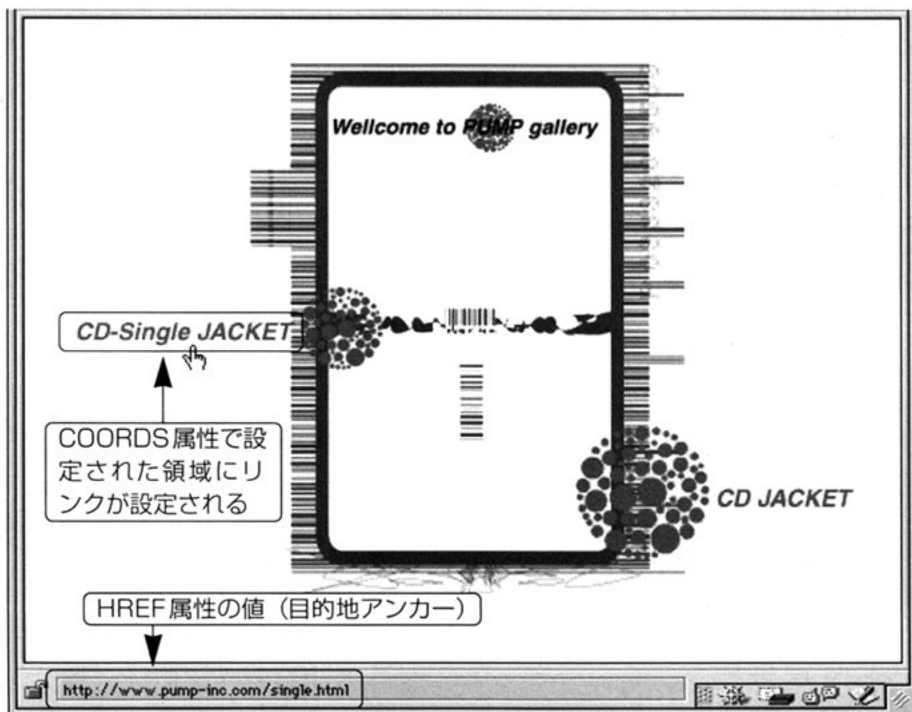
入手先　：**http://www.boutell.com/mapedit/**

●HREF属性

COORDS属性で設定された起点アンカー領域のリンク先リソースのURLを設定します。

```
<P ALIGN="center">
<IMG SRC="cd_index.gif" USEMAP="#cd_index"
BORDER="0">
<MAP NAME="cd_index">
<AREA SHAPE="rect" COORDS="20,160,220,210" HREF=
"single.html">
<AREA SHAPE="rect" COORDS="330,240,510,340"
HREF="cd.html">
</MAP>
</P>
```

※SHAPE・COORDS・HREF属性以外の属性は省略



AREA要素の使用例

マルチメディア

●NOHREF属性

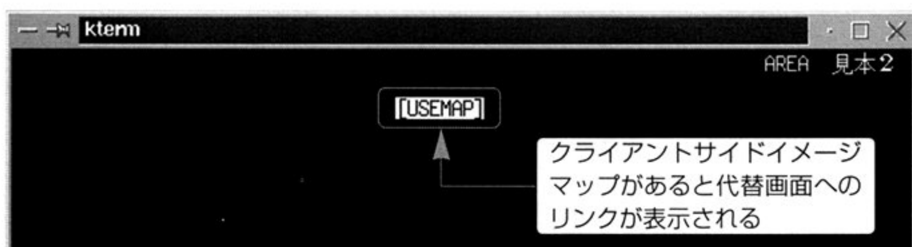
NOHREF属性を指定することによって、COORDS属性で設定された領域のリンク機能を無効にすることができます。先に設定された領域の中に、一部リンク無効の領域を設定する場合などに使用されます(たとえば、ドーナツリング状の領域の設定)。NOHREF属性を指定する場合は、属性名は省略し属性値「nohref」のみを記述します(P.8参照)。

●ALT属性

HREF属性で設定されたリンクの内容を知らせるための属性です。AREA要素では、必須の属性です。

現在では、Lynxなどテキストブラウザもクライアントサイド・イメージマップを使用できます。もちろん、イメージマップ(画像)そのものは表示できませんが、HREF属性の値を表示し、リンクとして使用することができるのです。ただ、HREF属性の値が生のまま表示されても、具体的にそのデータが何であるかは把握できないことが多いでしょう。そこで、ALT属性を設定し、HREF属性値の代替テキストとして表示させるのです。

```
<P ALIGN="center">
<IMG SRC="cd_index.gif" USEMAP="#cd_index" BORDER="0"
ALT="">
<MAP NAME="cd_index">
<AREA SHAPE="rect" COORDS="20,160,220,210" HREF=
"single.html" ALT="CD-Single JACKET">
<AREA SHAPE="rect" COORDS="330,240,510,340"
HREF="cd.html" ALT="CD JACKET">
</MAP>
</P>
```



Lynxでの表示例



イメージマップの代替画面を表示してくれる

注意 Webブラウザやアプリケーション（携帯端末用のものなど）の中には、クライアントサイド・イメージマップを全くサポートしていないものもあります。その場合は、ALT属性も意味をなしません。そのため、イメージマップ以外に、別途リンクを用意しておくことも必要です。

```
<P ALIGN="center">
<IMG SRC="cd_index.gif" USEMAP="#cd_index" BORDER="0"
ALT="">
<MAP NAME="cd_index">
<AREA SHAPE="rect" COORDS="20,160,220,210" HREF=
"single.html" ALT="CD-Single JACKET">
<AREA SHAPE="rect" COORDS="330,240,510,340"
HREF="cd.html" ALT="CD JACKET">
</MAP>
</P>

<P ALIGN="center">
<A HREF="single.html">［CD-Single JACKET］</A>
<A HREF="cd.html">［CD JACKET］</A>
</P>
```

イメージマップの代替データ

NN 3 IE 3 NN 4 IE 4

APPLET

<APPLET>~</APPLET>

Java Applet（Java APPLETs） テキストレベル要素

内容 PARAM
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 BODY
H1 H2 H3 H4 H5 H6 ADDRESS
P DIV CENTER BLOCKQUOTE FORM CAPTION
TH TD DT DD LI
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
APPLET FONT

要素の概要

APPLET要素は、HTML文書内でJava Appletを実行させるための要素です。

実行させるJava Appletや条件は各種属性で設定します。また、開始タグ<APPLET>と終了タグ</APPLET>によって、Java Appletの実行に必要なパラメータであるPARAM要素やJava Applet未対応のWebブラウザで表示すべき代替テキストをマークアップします。

なお、APPLET要素はテキストレベル要素なので、ブロックレベル要素やテキストレベル要素、あるいは一般的なテキストの中に置くことができます。また、Javaに関しては別途専門の解説書をご覧ください。

属性一覧

●CODEBASE（初期値：当該HTML文書の基準URL）

値	働き	H3	N4	I4
url	Java Appletの基準URL	○	○	○

属性は省略可

●CODE

値	働き	H3	N4	I4
code	JavaAppletのクラスファイル名	○	○	○

属性は必須

●ALT

値	働き	H3	N4	I4
text	代替テキスト	○	−	−

属性は省略可

●NAME

値	働き	H3	N4	I4
text	受信後のJava Appletに名前を設定	○	○	○

属性は省略可

●WIDTH

値	働き	H3	N4	I4
pixels	Java Appletの初期表示の横幅	○	○	○

属性は必須

●HEIGHT

値	働き	H3	N4	I4
pixels	Java Appletの初期表示の高さ	○	○	○

属性は必須

●ALIGN（初期値：bottom）

値	働き	H3	N4	I4
top	テキスト行上端にJava Appletの上端を揃えて配置	○	○	○
middle	テキスト行下端にJava Appletの中央を揃えて配置	○	○	○
bottom	テキスト行下端にJava Appletの下端を揃えて配置	○	○	○
left	テキスト行の左端にJava Appletを配置し、回り込みを設定	○	○	○
right	テキスト行の右端にJava Appletを配置し、回り込みを設定	○	○	○

属性は省略可

●HSPACE

値	働き	H3	N4	I4
pixels	Java Applet表示領域の左右の余白	○	○	○

属性は省略可

●VSPACE

値	働き	H3	N4	I4
pixels	Java Applet表示領域の上下の余白	○	○	○

属性は省略可

変数　*url*＝URL　*code*＝クラスファイル名（一般的なテキスト）
pixels＝数値　*text*＝一般的なテキスト

属性解説

●CODEBASE属性

Java Appletが置かれているURLを設定します。

属性を省略した場合は、当該HTML文書の基準URI（BASE要素）が使用されます（基準URLが設定されていない場合は、当該HTML文書のURLが使用される）。

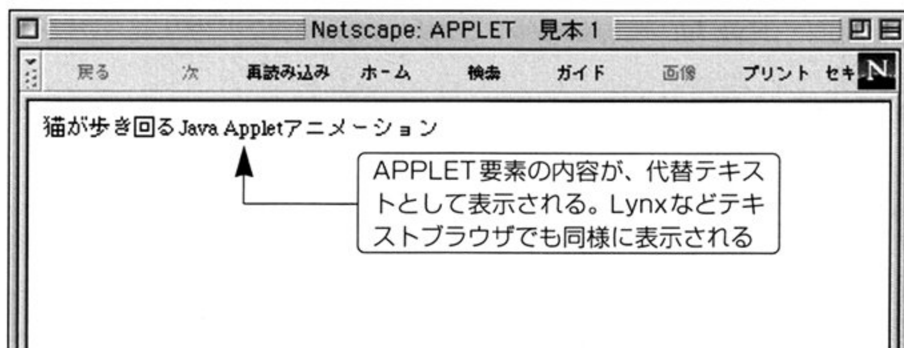
●CODE属性

実行させるJava Appletのクラスファイルを指定します。

```
<APPLET CODE="cat.class" WIDTH="200" HEIGHT="200">
猫が歩き回るJava Appletアニメーション
</APPLET>
```



Java Applet対応Webブラウザでの表示例



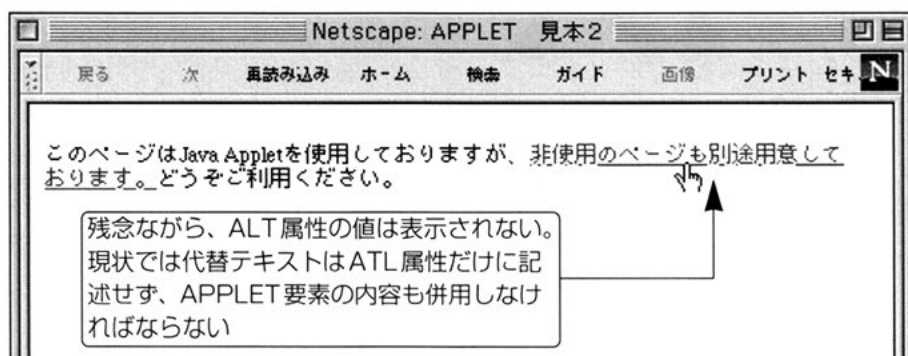
Java Applet対応WebブラウザでJava機能をオフにした場合の表示例

●ALT属性

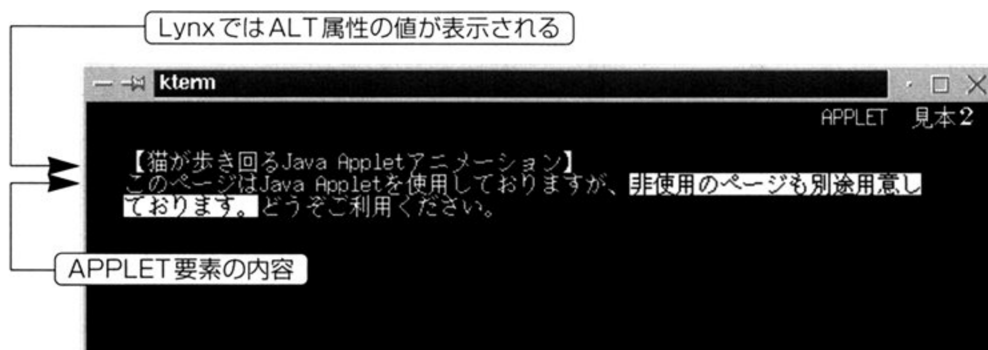
ALT属性の値は、WebブラウザなどのアプリケーションがAPPLET要素を認識できても、Java Appletを実行できない場合に表示される代替テキストです。一般的なWebブラウザはJava機能に未対応であっても、あるいは機能をオフにしていてもこの属性を無視するようですが、Lynxでは値が表示されます。

なお、Java Appletをアクセサリー程度に使用するだけなら代替テキストを記述するだけでいいのですが、リンクボタンなどWebページの制御用に使用する場合などは、別途Java Applet不要のページも用意しなければなりません。

```
<P>
<APPLET CODE="cat.class" WIDTH="200" HEIGHT="200"
ALT="【猫が歩き回るJava Appletアニメーション】"><BR>
このページはJava Appletを使用しておりますが、<A HREF=
"main_njava.html">非使用のページも別途用意しております。</A>ど
うぞご利用ください。
</APPLET>
</P>
```



Java Applet対応WebブラウザでJava機能をオフにした場合の表示例



Lynxでの表示例

●NAME属性

Webブラウザなどが Java Appletを受信し、Webページ上で実行する際に使用する名前を設定します。同じHTML文書内で複数のJava Appletを動作させ、それらがデータのやり取りをする場合、この名前でお互いを認識します。

●WIDTH属性

Java Appletが最初に実行される際の領域の横幅をpixel数で設定します。IMG

要素などと異なり、省略することはできません。

なお、Java Appletが実行されることによって開くウインドウやダイアログボックスなどの大きさはこの設定に影響されません。

●HEIGHT属性

Java Appletが最初に実行される際の領域の高さをpixel数で設定します。IMG要素などと異なり、省略することはできません。

なお、Java Appletが実行されることによって開くウインドウやダイアログボックスなどの大きさはこの設定に影響されません。

●ALIGN属性

Java Appletの動作領域とJava Appletが含まれるテキスト行との位置関係を設定します。ただし、Webブラウザや表示フォントによっては、表現結果が微妙に異なる場合があります。

属性値top、middle、bottomは、テキスト行に対して画像を揃える位置を設定します。また、ALIGN属性を省略すると、初期値としてbottomが使用されます。

表示結果はIMG要素のALIGN属性を参照してください(P.221参照)。

●HSPACE属性

Java Appletの左右の余白をpixel数で設定します。属性を省略すると、0以外の最小値が設定されます（具体的な値はWebブラウザによって異なる)。

表示結果はIMG要素のHSPACE属性を参照してください(P.225参照)。

●VSPACE属性

Java Appletの上下の余白をpixel数で設定します。属性を省略すると、0以外の最小値が設定されます(具体的な値はWebブラウザによって異なる)。

PARAM

<PARAM>（終了タグなし）

Java Appletのパラメータ（The PARAM element）　テキストレベル要素

内容 ▶ 空

親要素 ▶ APPLET

要素の概要

PARAM要素は、APPLET要素で設定されたJava Appletの実行に必要なパラメータを設定するための要素です。必ずAPPLET要素の中の、それも最初に置かれます。

なお、PARAM要素は開始タグ<PARAM>のみでマークアップされる、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

属性一覧

●NAME

値	働き	H3	N4	I4
text	パラメータの名前を設定	○	○	○

属性は必須

●VALUE

値	働き	H3	N4	I4
text	パラメータの値	○	○	○

属性は省略可



text＝一般的なテキスト

属性解説

●NAME属性

Java Appletが実行時に必要とするパラメータの名前を設定します。PARAM要素必須の属性です。

●VALUE属性

パラメータの値を設定します。

```
<P>
<APPLET CODE="cat.class" WIDTH="200" HEIGHT="200"
ALT="【猫が歩き回るJava Appletアニメーション】"><PARAM
NAME="sound" VALUE="footfall.aiff|mew.aiff"><BR>
このページはJava Appletを使用しておりますが、<A HREF=
"main_njava.html">非使用のページも別途用意しております。</A>ど
うぞご利用ください。
</APPLET>
</P>
```

Column　プラグインよりJava？

現在インターネット上には、表示にShockwaveなどプラグイン（P.290参照）を必要とするWebページが数多くあります。このプラグインは、Webブラウザそのものの機能を拡張してくれる便利なものですが、使用するためにはあらかじめインストールしておかなければなりません。しかも、使用するデータに合わせて何種類も用意しなければなりません。

しかし、データ制作者がプラグインに相当するプログラムをJavaで作成して、データと一緒に配信すれば、この問題はかなり解決できます。なぜなら、Javaのプログラムは、基本的にどのJavaVM（Java Virtual Machine：Javaプログラムを実行するためにソフトウェアで構築された仮想コンピュータ）でも動作するので、ユーザはJavaVMを1つ用意しておけば、それ以上の手間は不要になるからです。あとはWebサイトへアクセスするとJavaのプログラムとデータが自動的にダウンロードされ、Webブラウザ（あるいはOS上）のJavaVM内で実行されます。

以前はJavaプログラムは1から作成していましたので、一般人にはちょっと敷居が高かったのですが、最近ではデータの作成時に、自動的に必要なJavaプログラムも作成してくれるソフトウェア（Macromedia社のDirectorやGEO社のEmblazeシリーズなど）もありますので、今後さらにJavaプログラムを利用したWebページが普及するでしょう。ただ、その前に各社のJavaVM間の互換性をなんとかして欲しいものです…。

拡張要素の注意点

これまで紹介してきたHTML 3.2の要素とは別に、Webブラウザメーカーによって考案されたさまざまな拡張要素が存在します。それらの中には、スクリプトやフレーム機能のように、HTML 4.0において採用されたものもあります。

HTML 4.0で採用された拡張要素

スクリプトとフレーム関連の要素は、Netscape Navigator 2.0やInternet Explorer 3.0に実装されたこともあって、古くから多くのWebページで使用されていますが、HTML 3.2では規格外の要素です。そこで、どうしてもフレームやスクリプトを使用したい場合は、HTML 3.2ではなくHTML 4.0の文書として記述する必要があります。

W3CのHTML 4.0には「Strict DTD」「Transitional DTD」「Frameset DTD」と呼ばれる3種類のDTDが用意されていますが(P.303参照)、普段HTML 3.2を使用している人が例外的にHTML 4.0を使用しなければならない場合は、Transitional DTDに従うといいでしょう。なぜなら、Transitional DTDはHTML 3.2との互換性を考慮した、いわゆる「移行用」の規格なので、以下の点を確認すれば従来のHTML文書がほぼそのまま使用できるからです。

●使用できなくなる要素の確認

以前から使用を控えるよう推奨されてきたLISTING・PLAINTEXT・XMPの3要素は、完全に破棄されました。

●必須になる属性の確認

HTML 3.2において使用が推奨されていたIMG要素のALT属性は、必須の属性となりました。

●制限が厳しくなる要素

TR要素の内容が0個以上から1個以上に変更されました。

いかがでしょう。この程度の変更であれば比較的容易にできると思います(P.258コラム参照)。もちろん、このような安直な手段で出来上がった文書は、

HTML 4.0特有の機能が使用されていませんから「HTML 4.0らしい文書」とはいえません。ただ、内容と合致しない文書型宣言をしてしまうよりはいいのではないかと思います。

なお、フレームの基本構造を設定するHTML文書は、Transitional DTDではなく、専用のFrameset DTDに従わなければなりません。もっとも、とりあえず使う場合に憶えなければならない規則はそれほど多くはありません。

Webブラウザメーカーの独自拡張要素

HTML 3.2や4.0とは別に、Webブラウザメーカーが独自に拡張した要素も現実には使用されています。それらのうち、比較的知られている要素について、最後にいくつか紹介しておきました。

ただし、これらの要素を使用した場合、HTML文書の文書型宣言が問題になります。当然ながら、HTML 3.2や4.0の文書型宣言を行ってはいけません。もちろん、メーカーが公開したDTDの文書型宣言を行えばいいのですが、困ったことに一番多くの独自拡張を行ってきたNetscape社はDTDを公開していません。また、Microsoft社にしても公開しているDTDは、Internet Explorer 3.0 β版用だけです。正直に申し上げて、独自拡張要素を使用した場合の文書型宣言は、どうしていいか筆者にはわかりません。使用される方のご判断にお任せします（Microsoft本社の解説ページでは、なぜかHTML 3.2 DTDとHTML 4.0 Strict DTDの宣言例が載っていますが…）。

なお、本書では独自拡張要素については主な機能だけを紹介しています。また、EMBED要素のように扱うプラグインによって要素の属性がさまざまに変化するものはとても紹介しきませんので、使用される場合は各プラグインメーカーが配布している資料をご覧になってください。

拡張

SCRIPT

<SCRIPT>～</SCRIPT>

スクリプト（The SCRIPT element）

内容 ▶ 一般的なテキスト（本文参照）

親要素 ▶ HEAD　BODY

要素の概要

SCRIPT要素は、使用するクライアントサイド・スクリプト言語（Webブラウザが解釈して実行するスクリプト言語。JavaScriptやVBScript、Tclなど）を宣言し、スクリプトを記述するために使用します。

開始タグ<SCRIPT>と終了タグ</SCRIPT>によって、スクリプトをマークアップします。また、SCRIPT要素の内容は他の要素の内容と異なり、空白文字の連続やタブ、「<」や「&」などの特殊な文字（P.310参照）はそのまま認識されます（うっかり文字実体参照で記述すると、それはただの記号として認識される）。そして、最初に「<」「/」の連続（終了タグの開始記号）が現れた時点で、内容の終了として認識されます。

なお、スクリプトを十分に活用するためには、HTML 4.0の解説書、および各スクリプト言語の解説書をご覧ください。

属性一覧

●TYPE

値	働き	H4	N4	I4
content-type	使用するスクリプトの種類	○	×	○

属性は必須

●LANGUAGE（廃止予定）

値	働き	H4	N4	I4
script	使用するスクリプトの種類	○	○	○

属性は必須（TYPE属性が有効な場合は不要）

●SRC

値	働き	H4	N4	I4
url	使用するスクリプトのURL	○	○	○

属性は省略可

●DEFER（論理型）

値	働き	H4	N4	I4
defer	スクリプトの実行を待たずにHTML本文を表示	○	×	○

属性は省略可

●CHARSET

値	働き	H4	N4	I4
charset	リンク先のリソースの文字の符号化方式	○	×	×

属性は省略可

変数 *content-type*＝スクリプトのContent-Type
script＝スクリプトの言語名
url＝URL　*charset*＝文字の符号化方式

属性解説

●TYPE属性

使用するスクリプト言語の種類をMIMEのContent-Type形式で記述します。HTML 4.0ではスクリプト言語の種類について、特に「HTML専用」というものを用意していませんので、スクリプトを使用する際には、必ずその種類を明示しなくてはなりません。したがって、この属性は必須です。

なお、Netscape社の独自規格では、スクリプト言語はLANGUAGE属性で設定していましたが、Transitional DTDではこちらを使用するよう推奨されています（Strict DTDではLANGUAGE属性は破棄されている）。

簡単なJavaScriptの例

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=iso-2022-jp">
<TITLE>JavaScript Test</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function goodmorning(){
window.alert("おはようございます！")
}
```

文書型宣言（HTML4.0 Transitional DTDに基づく）
コメントの開始
開始タグ 要素の内容がJavaScriptであることを明示
JavaScript本文

拡張

```
//-->                              ← コメントの終了
</SCRIPT>                          ← 終了タグ
</HEAD>
<BODY onLoad="goodmorning()">      ← イベントの設定
<P>onLoad属性で設定されたスクリプトは、HTML文書の読み込み完了をき
っかけとして実行されます。</P>
(以下略)
```

注意 SCRIPT要素に対応したWebブラウザは、SCRIPT要素の内容を画面に表示せずスクリプトとして認識・実行しますが、SCRIPT要素に未対応のWebブラウザではそのまま内容を表示してしまいます。それを防ぐため、SCRIPT要素の内容は「コメント」として記述します。



JavaScriptの実行例

●LANGUAGE属性

使用するスクリプト言語名を記述します。Transitional DTDではTYPE属性を使用するよう推奨されています(Strict DTDではLANGUAGE属性は破棄されている)。

ただし、Netscape NavigatorはTYPE属性に対応していませんので、しばらくはLANGUAGE属性も併用することになるでしょう。

```
<SCRIPT LANGUAGE="javascript">
```

TYPE属性と異なり、言語名だけを記述

●SRC属性

使用するスクリプトを当該HTML文書内に記述せず、別のファイルに記述して、それを呼び出して使用することもできます。その場合、SRC属性でスクリプトを記述したファイルのURLを設定します。

●DEFER属性

DEFER属性を指定することによって、スクリプトがHTML文書表示の際にテキストを挿入しないことを明示します。そのため、Webブラウザはスクリプトの実行終了を待たずに、HTML文書を表示し始めることができます。

DEFER属性を指定する場合は属性名は省略し、属性値「defer」のみを記述します。

●CHARSET属性

リンク先のリソースの文字の符号化方式を設定します。

NN 3 | IE 3 | NN 4 | IE 4

NOSCRIPT <NOSCRIPT>~</NOSCRIPT>

スクリプト未対応Webブラウザ用の代替要素（The NOSCRIPT element）　ブロックレベル要素

内容 H1 H2 H3 H4 H5 H6
ADDRESS
P UL OL DL (DIR) (MENU) PRE DIV CENTER
BLOCKQUOTE FORM ISINDEX HR TABLE
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

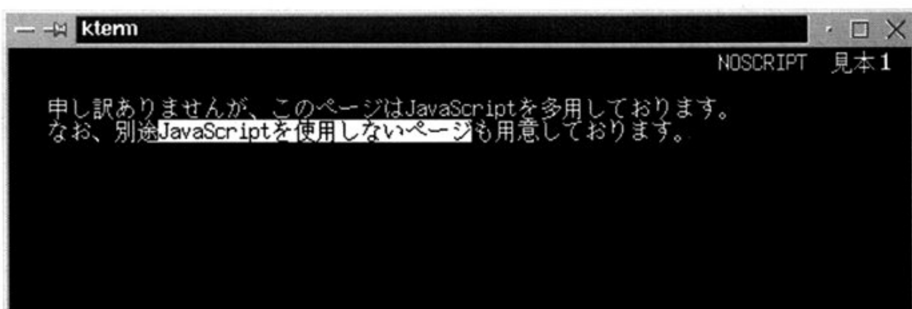
親要素 BODY
DIV CENTER BLOCKQUOTE FORM TH TD

要素の概要

NOSCRIPT要素は、スクリプトに未対応、あるいはスクリプト機能をオフにしているWebブラウザやアプリケーションのための代替要素です。反対に、スクリプトに対応しているWebブラウザでは内容は表示されません。

開始タグ<NOSCRIPT>と終了タグ</NOSCRIPT>によって、代替テキストなどをマークアップします。一般的には、リンクなどを設定して、スクリプト機能が不要なページなどを案内します。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transi
tional//EN">
（中略）
<SCRIPT TYPE="text/javascript">
（スクリプト省略）
</SCRIPT>
<NOSCRIPT>
<P>
申し訳ありませんが、このページはJavaScriptを多用しております。<BR>
なお、別途<A HREF="http://www.pump-inc.com/nscript_in
dex.html">JavaScriptを使用しないページ</A>も用意しております。
</P>
</NOSCRIPT>
（以下略）
```



Lynxでの表示例

属性一覧

省略

属性解説

省略

注意 NOSCRIPT要素には、HTML 4.0の基本属性の多くが設定できますが、それらについては別途HTML 4.0の解説書をご覧ください。

Column　NOSCRIPT要素の機能について

「NOSCRIPT要素は、スクリプト未対応のWebブラウザへの配慮」と本文では書いていますが、「スクリプト未対応のWebブラウザなら、<NOSCRIPT></NOSCRIPT>というタグも認識できないのでは？」という疑問が沸くかもしれません。これはその通りで、実際にスクリプト未対応のWebブラウザは<SCRIPT>も<NOSCRIPT>も認識できません。要するにNOSCRIPT要素の内容だけを解釈して表示いるのです。

反対に、スクリプトに対応したWebブラウザは、<NOSCRIPT></NOSCRIPT>タグを認識して、その内容を表示（解釈）しないようにします。

なお、フレーム未対応のWebブラウザのためのNOFRAMES要素やプラグイン未対応WebブラウザのためのNOEMBED要素についても、その機能は同様です。

拡張

NN 3 IE 3 NN 4 IE 4

FRAMESET <FRAMESET>～</FRAMESET>

フレームの設定（The FRAMESET element）　BODY要素に相当

内容 ▶ FRAMESET
FRAME　NOFRAMES

親要素 ▶ HTML

要素の概要

フレームの基本構造は、通常のHTML文書とは別に専用のHTML文書で設定します。FRAMESET要素は、その専用HTML文書に置かれ、フレームの基本構造を設定するために使用します。

FRAMESET要素は、通常のHTML文書のBODY要素に相当するもので、その内容には実際にフレーム内に表示するリソース（HTML文書に限りません）を設定するFRAME要素、フレーム未対応のWebブラウザのための代替要素であるNOFRAMES要素が置かれます。また、FRAMESET要素は入れ子にすることができますので、複雑なフレームを設定することも可能です（実用的かどうかは別問題）。

なお、FRAMESET要素の置かれる「フレーム設定用のHTML文書」は、専用のFrameset DTDに基づいて作成しなければなりません。そして、作成された文書は以下のような文書型宣言を行います。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Frameset//EN">
```

一方、フレーム設定用のHTML文書によって呼び出される（フレーム内に表示される）HTML文書は、Transitional DTDに基づいて作成しなければなりません（HTML 3.2の文書はそのままでは使用できません）。そして、作成された文書は以下のような文書型宣言を行います。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transi
tional//EN">
```

注意 フレームは便利な機能と思われていますが、未対応のWebブラウザやアプリケーションのユーザにとっては大変やっかいな存在です。また、フレーム対応Webブラウザのユーザの中にも、フレームが嫌いなユーザは大勢います。たとえば、画面の狭いノートパソコンなどを使用していると、フレームがあると表示面積が狭められ、横スクロールしなければ内容が見渡せなくなります。あるいは、フレームが枠とスクロールバーで埋まってしまい、文字がさっぱり読めない時もあります。
フレームを使用する前に、本当に自分の文書はフレームが必要なのかどうか、今一度検討してください。また、どうしてもフレームを使用したい場合は、NOFRAMES要素（P.270参照）も必ず併用してください。

属性一覧

●ROWS（初期値：100%）

値	働き	H4	N4	I4
multi-length	横方向の分割領域設定	○	○	○

属性は省略可

●COLS（初期値：100%）

値	働き	H4	N4	I4
multi-length	縦方向の分割領域設定	○	○	○

属性は省略可

変数 *multi-length*＝pixel数、%値、相対値

注意 以上の他にも、HTML 4.0の基本属性の多くが設定できますが、それらについては別途HTML 4.0の解説書をご覧ください。

属性解説

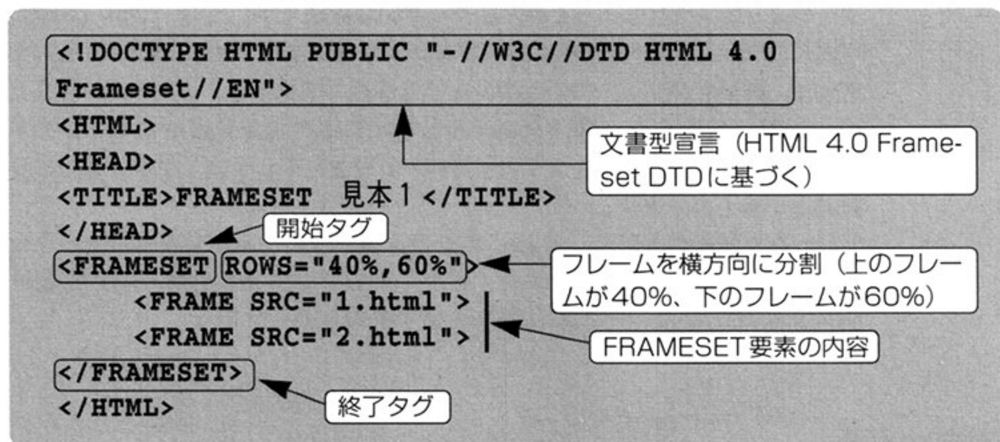
●ROWS属性

フレームの横方向の分割設定を行います。

pixel数による絶対値、ウインドウの高さに対する%値、あるいは相対値（数値の後ろに「*」を付加）のいずれかの値により、各フレームの高さを設定します。また、それらは設定するフレームの数に合わせて、コンマで区切って記述します（設定方向は上から下の順になる）。

なお、属性値を省略すると100%が設定されます。この場合は、横方向の分割は

行われません。



※ TITLE要素以外のHEAD要素、及びNOFRAMES要素は省略



ROWS属性の設定例（横に2分割した場合）

●COLS属性

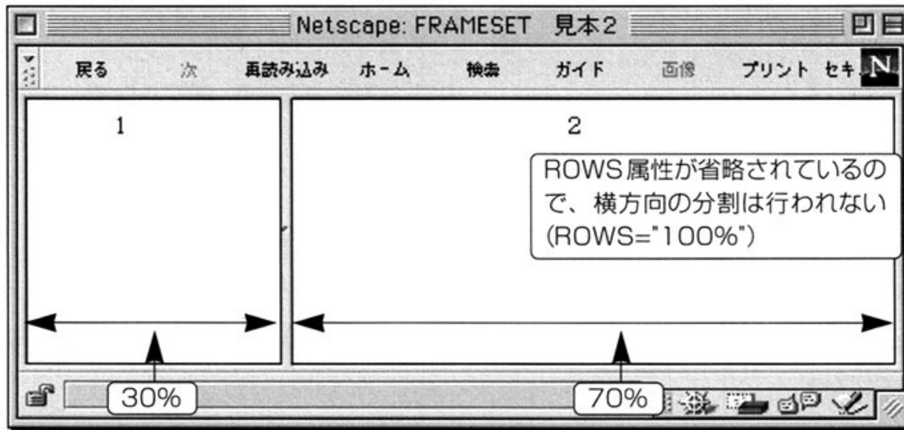
フレームの縦方向の分割設定を行います。

pixel数による絶対値、ウインドウの高さに対する%値、あるいは相対値のいずれかの値により、各フレームの横幅を設定します。また、それらは設定するフレームの数に合わせて、コンマで区切って記述します（設定方向は左から右の順になる）。

なお、属性値を省略すると100%が設定されます。この場合は、縦方向の分割は行われません。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frame
set//EN">
(中略)
<FRAMESET COLS="30%,70%">
     <FRAME SRC="1.html">
     <FRAME SRC="2.html">
</FRAMESET>
```

フレームを縦方向に分割（左のフレームが30%、右のフレームが70%）



COLS属性の設定例（縦に2分割した場合）

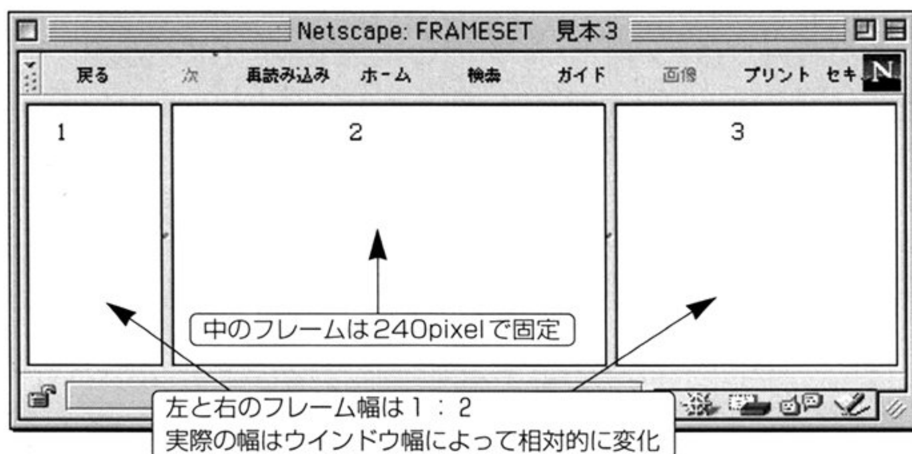
設定値の混在

pixel数による絶対値、ウインドウの高さに対する%値、相対値による各設定方法は、同時に使用することができます。たとえば、3つのフレームを設定し1つは絶対値によって固定し、残り2つを相対値によって適当に設定することもできます。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frame
set//EN">
(中略)
<FRAMESET COLS="1*,240,2*">
     <FRAME SRC="1.html">
     <FRAME SRC="2.html">
     <FRAME SRC="3.html">
</FRAMESET>
```

フレームを縦方向に分割（中のフレームの横幅が240pixel、残りを左右のフレームで1対2の比率で分割）

拡張



COLS属性の設定例（縦に3分割した場合）

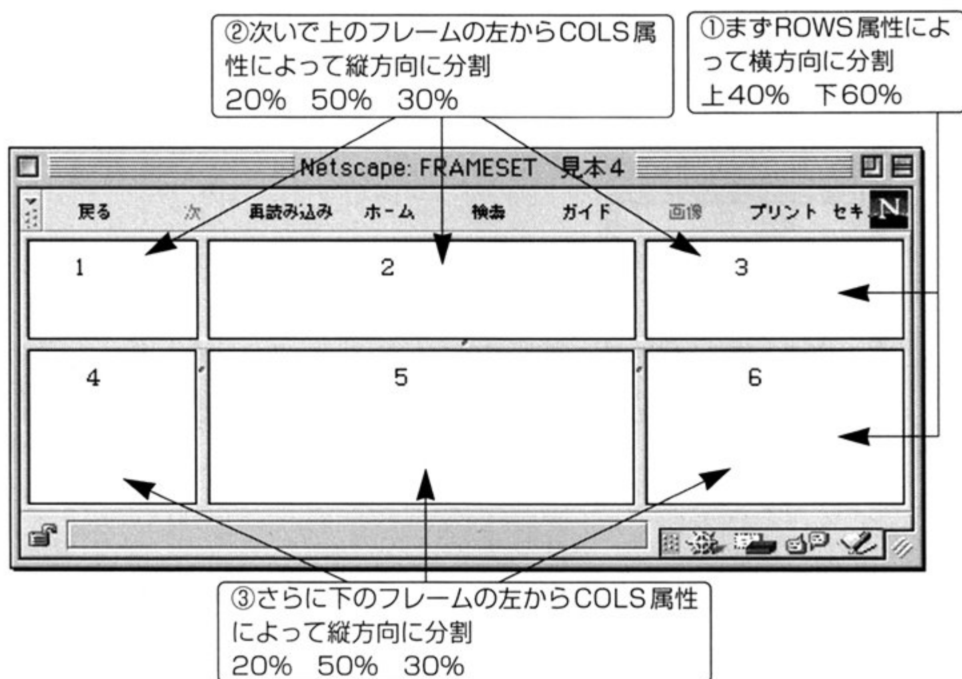
ROWS属性とCOLS属性の混在

ROWS属性とCOLS属性は同時に使用することができます。その場合は、まずROWS属性によって横方向の分割が行われ、次いで上のフレームの左から右へCOLS属性によって縦方向の分割が行われ、さらに下のフレームも順次左から右へ分割されます。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frame
set//EN">
(中略)
<FRAMESET ROWS="40%,60%" COLS="20%,50%,30%">
     <FRAME SRC="1.html">
     <FRAME SRC="2.html">
     <FRAME SRC="3.html">
     <FRAME SRC="4.html">
     <FRAME SRC="5.html">
     <FRAME SRC="6.html">
</FRAMESET>
```

ROWS属性によって横方向に分割
(40%　60%)

CLOS属性によって縦方向に分割
(20%　50%　30%)



COLS属性の設定例（縦3分割　横に2分割した場合）

フレームの入れ子

FRAMESET要素の中に別のFRAMESET要素を置くと、フレームを入れ子にすることができます。また、その数に制限はありません(実用性は別の問題)。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frame
set//EN">
(中略)
<FRAMESET COLS="30%,70%">
    <FRAMESET ROWS="40%,60%">
        <FRAME SRC="1.html">
        <FRAME SRC="2.html">
    </FRAMESET>
    <FRAME SRC="3.html">
</FRAMESET>
```

COLS属性によって縦方向に分割（30%　70%）

左のフレームをROWS属性によって横方向に分割（40%　60%）

拡張

フレームの入れ子の例

Column HTML 3.2とHTML 4.0の違い（補足）

HTML 3.2とHTML 4.0（Transitional DTD）の違いについて、細かいことをいうともう1点違いがあります。HTML 3.2ではTABLE要素の中に行データであるTR要素が置かれていましたが、HTML 4.0では、表の内容をグループ化するTHEAD（表のヘッダ）・TFOOT（表のフッタ）・TBODY（表の本体、必須）の3種類の要素が置かれ、その中にTR要素が置かれます。これによって、表のデータを適切にグループ化し、データをスクロールさせたり、的確な音声出力を実現することができます。

```
<TABLE>  ← 表組の開始
<THEAD>
<TR><TH> A </TH><TH> B </TH><TH> C </TH><TH> D </TH></TR>
</THEAD>
<TBODY>
<TR><TD> 1 </TD><TD> 2 </TD><TD> 3 </TD><TD> 4
</TD></TR>
</TBODY>
</TABLE>  ← 表組の終了
```


では、HTML 3.2 DTDに基づいて作成した表は変更が必要なのでしょうか？　いえ、これはそのままで大丈夫です。というのも「THEAD・TFOOT要素が存在せず、TBODY要素が1つしか存在しない場合は、開始タグ<TBODY>は省略可能」なのです（終了タグ</TBODY>は常に省略可能）。HTML 3.2 DTDに基づいて作成された表には、当然ながらTHEAD・TFOOT要素共に存在しませんし、TBODY要素に相当する表の本体も1つしか存在していません。したがって、文書的には変更は必要ないのです。

NN 3 IE 3 NN 4 IE 4

FRAME

<FRAME>（終了タグなし）

フレーム内の設定（The FRAME element）

内容 ▶ 空

親要素 ▶ FRAMESET

要素の概要

FRAME要素は、フレーム内に表示される具体的なデータです。といっても、実際にはデータそのものではなく、画像やJava Appletなどと同様、SRC属性によってデータとなるHTML文書（リソース）のURL先を設定します。また、余白やスクロール、フレーム枠の形状などもFRAME要素の属性で設定することができます。

なお、FRAME要素は開始タグ<FRAME>のみでマークアップされる、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

注意 FRAME要素のSRC属性によってリンクされる（フレーム内に表示される）HTML文書の内容は、HTML 3.2ではなくTransitional DTDに従わなければなりません（HTML 3.2からの変更点についてはP.244参照）。また、文書型宣言も以下のように変更します。
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

属性一覧

●SRC

値	働き	H4	N4	I4
url	フレーム内に表示するデータのURL	○	○	○

属性は省略可

●NAME

値	働き	H4	N4	I4
text	ターゲットにするフレーム名の設定	○	○	○

属性は省略可

●LONGDESC

値	働き	H4	N4	I4
url	フレームの内容に関する解説データのURL	○	×	×

属性は省略可

●NORESIZE（論理型）

値	働き	H4	N4	I4
noresize	フレームサイズの固定	○	○	○

属性は省略可

●SCROLLING（初期値：auto）

値	働き	H4	N4	I4
yes	スクロールバーを常時表示	○	○	○
no	スクロールバーを常時非表示	○	○	○
auto	必要な場合にスクロールバーを表示	○	○	○

属性は省略可

●FRAMEBORDER（初期値：0）

値	働き	H4	N4	I4
1	フレーム枠を表示	○	○	○
0	フレーム枠を非表示	○	○	○

属性は省略可

●MARGINWIDTH

値	働き	H4	N4	I4
pixels	フレーム内の左右の余白	○	○	○

属性は省略可

●MARGINHEIGHT

値	働き	H4	N4	I4
pixels	フレーム内の上下の余白	○	○	○

属性は省略可

変数 *url*＝URL　*text*＝一般的なテキスト
pixels＝pixel数

注意 以上の他にも、HTML 4.0の基本属性の多くが設定できますが、それらについては別途HTML 4.0の解説書をご覧ください。

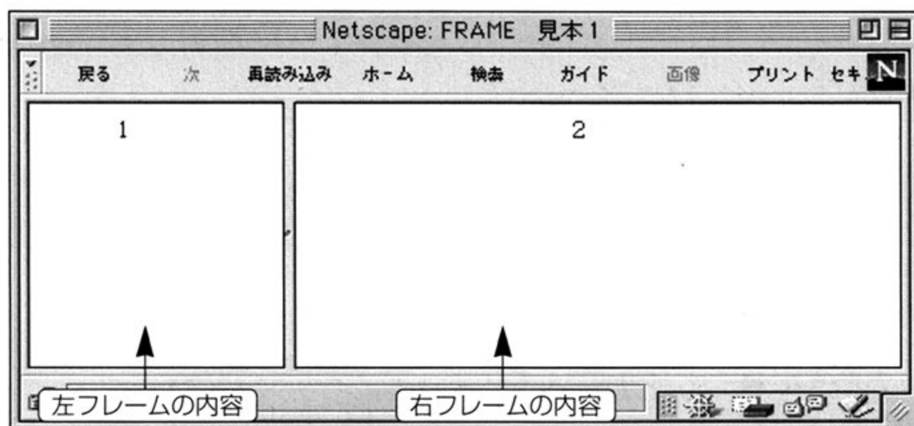
属性解説

●SRC属性

フレーム内に表示させるデータを含むHTML文書のURLを設定します。また、

HTML文書以外のリソース(画像など)を直接設定することもできます。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frame
set//EN">
(中略)
<FRAMESET COLS="30%,70%">
    <FRAME SRC="1.html">   ← 左フレームの内容
    <FRAME SRC="2.html">   ← 右フレームの内容
</FRAMESET>
```



FRAME要素の使用例

●NAME属性

ターゲットフレーム名を設定します。この値は、同時にLINK・A・AREA・FORM要素のTARGET属性(HTML 4.0で追加された属性)の値として使用されます。

たとえば、左フレームにリンク(起点アンカー)を設定した目次や索引を表示し、右側のフレームにはそのリンク先リソースを表示する場合などに使用されます。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frame
set//EN">
(中略)
<FRAMESET COLS="40%,60%">
    <FRAME SRC="ad_index.html">
    <FRAME SRC="default.html" NAME="water_ad">
</FRAMESET>
```

左フレームに表示するリソースの指定（ad_index.html）
右フレームに設定するターゲットフレーム名（water_ad）
右フレームに表示するリソース指定（default.html）

フレームの基本構造の設置

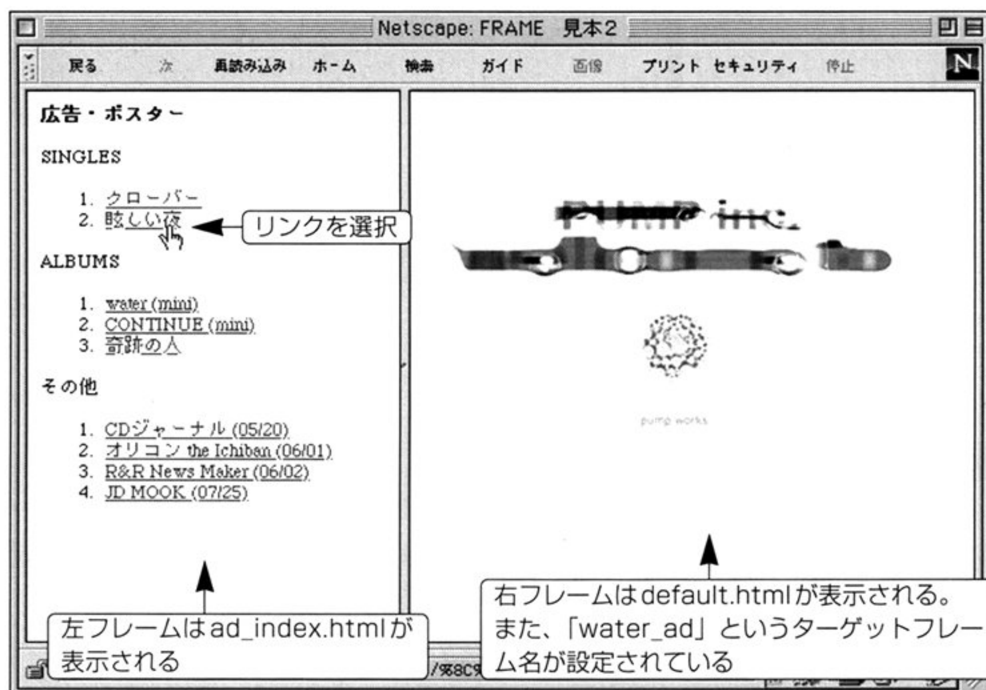
拡張

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transi
tional//EN">
(中略)
<OL>
<LI><A HREF="clover_p.html" TARGET="water_ad">クローバ
ー</A></LI>
<LI><A HREF="mabushiiyoru_p.html" TARGET="water_ad">
眩しい夜</A></LI>
</OL>
```

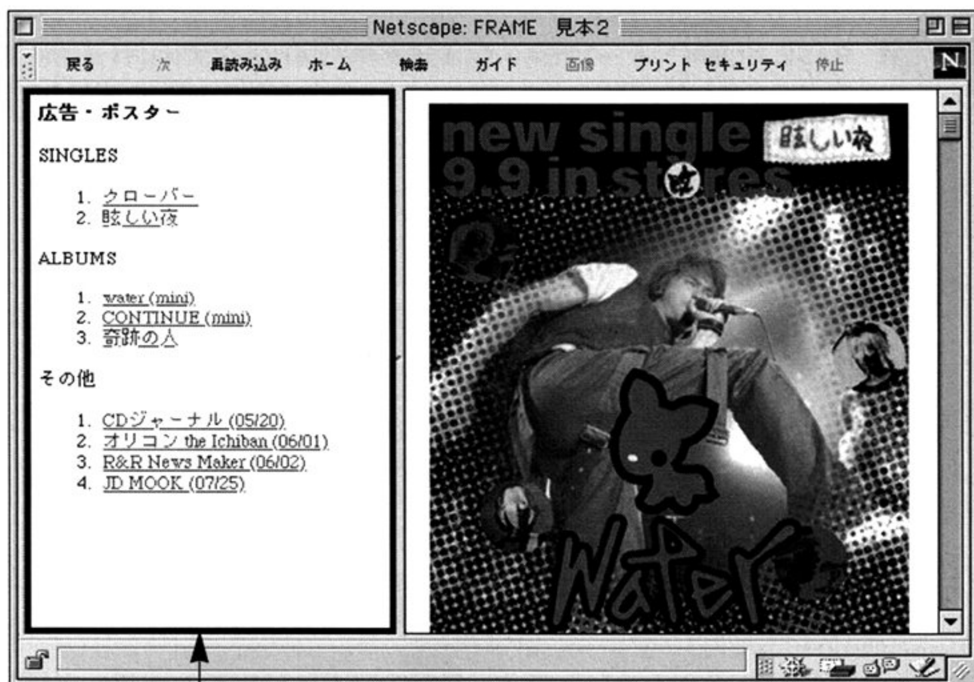
リンク先のHTML文書「clover_p.html」は「water_ad」という名のフレームに表示

リンク先のHTML文書「mabushiiyoro_p.html」は「water_ad」という名のフレームに表示

左フレームに表示する「Tad_index.html」
※HREF・TARGET属性以外の属性は省略



①フレームの最初の表示



左フレームはそのまま

②右フレーム（A要素のTARGET属性で設定された「water_ad」）にリンク先リソース（mabushiiyoru_p.html）が表示される

なお、BASE要素のTARGET属性（HTML 4.0で追加された属性）で基準となるターゲットフレーム名を設定してある場合は、個々のLINK・A・AREA・FORM要素でTARGET属性を設定する必要はありません（設定した場合はBASE要素のTARGET属性より優先される）。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transi
tional//EN">
(中略)
<BASE TARGET="water_ad">
(中略)
<OL>
<LI><A HREF="clover_p.html">クローバー</A></LI>
<LI><A HREF="mabushiiyoru_p.html">眩しい夜</A></LI>
</OL>
```

この文書全体に有効となる基準ターゲットフレーム名

個々のA要素でTARGET属性を設定する必要がなくなる

※HREF・TARGET属性以外の属性は省略

拡張

注意 BASE・LINK・A・AREA・FORM要素のTARGET属性にはHTML文書作成者が自由に指定するターゲットフレーム名の他に、特別な値がいくつか用意されています。

・_blank
　新規にウインドウを開き、リンク先のリソースを表示します。
・_self
　リンク（起点アンカー）を設定したHTML文書と同じフレームにリンク先のリソースを表示します。
・_parent
　リンク（起点アンカー）を設定したHTML文書と同じフレームの親フレームにリンク先のリソースを表示します。特に親フレームがない場合は「_self」と同じ結果になります。
・_top
　リンク（起点アンカー）を設定したHTML文書が表示されていたフレームを解除し、そのウインドウ内にリンク先のリソースを表示します。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frame
set//EN">
(中略)
<FRAMESET COLS="40%,60%">
     <FRAME SRC="cd_index.html">
     <FRAME SRC="default.html" NAME="water_cd">
</FRAMESET>
```


フレームの基本構造の設定

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transi
tional//EN">
(中略)
<OL>
<LI><A HREF="clover_cd.html" TARGET="_blank">クローバー
</A></LI>
<LI><A HREF="mabushiiyoru_cd.html" TARGET="_self">眩
しい夜</A></LI>
</OL>
```

左フレームに表示する「od-index.html」
※HREF・TARGET属性以外の属性は省略



TARGET="_blank"の表示例



TARGET="_self"の表示例

拡張

●LONGDESC属性

このフレーム内のHTML文書に関する解説データを別のHTML文書に持たせる場合、そのURLを設定します。たとえば、音声出力を利用する場合などに役立ちます。

●NORESIZE属性

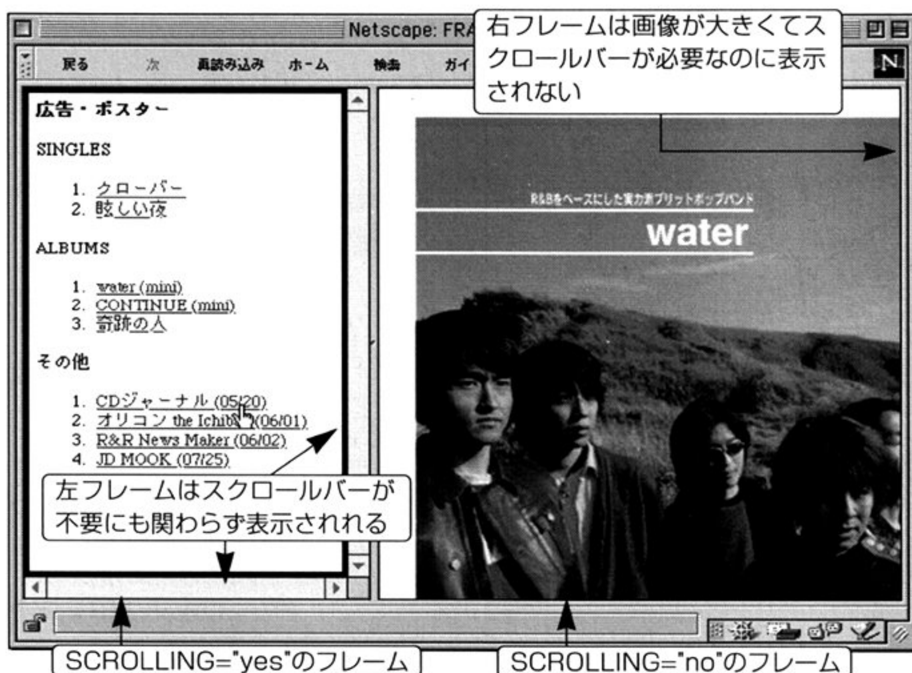
多くのWebブラウザでは、フレーム枠のサイズ変更はユーザが自由に行えます。しかし、NORESIZE属性を設定することによって、フレーム枠のサイズを固定してしまうことができます。NORESIZE属性を設定する場合は、属性名は省略し属性値「noresize」のみを記述します（P.8参照）。

●SCROLLING属性

SCROLLING属性を設定することによって、フレーム内のスクロールバーの表示／非表示を制御することができます。

属性値に「yes」を設定すると、スクロールバーは常時表示されます。また「no」を設定するとスクロールバーは表示されません。あるいは「auto」を設定すると、フレーム内のデータがフレームより大きい場合にのみ、スクロールバーが表示されます。なお、属性を省略した場合は、自動的に「auto」が設定されます。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Frameset//EN">
(中略)
<FRAMESET COLS="40%,60%">
     <FRAME SRC="ad_index.html" SCROLLING="yes">
     <FRAME SRC="default.html" NAME="water_ad"
SCROLLING="no">
</FRAMESET>
```



SCROLLING属性の使用例

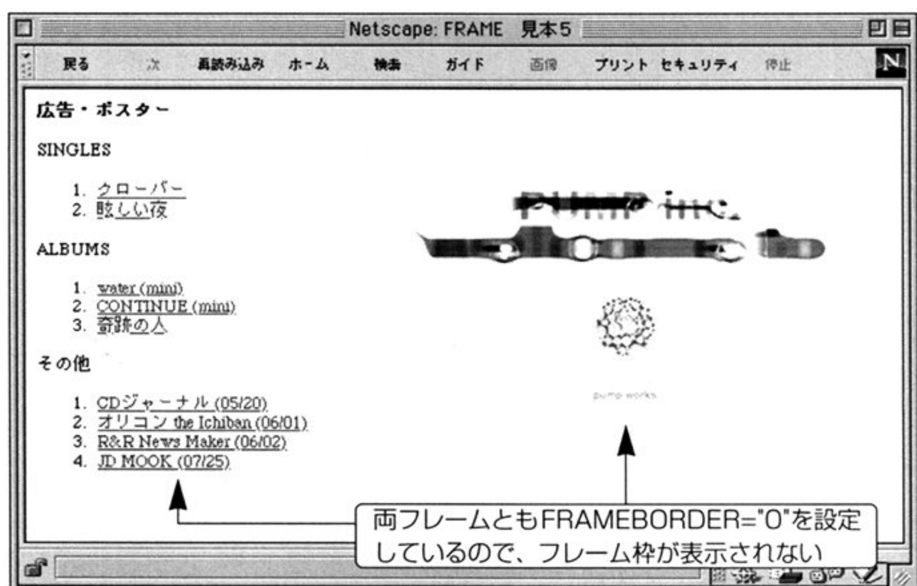
●FRAMEBORDER属性

フレーム枠（隣接するフレームとの区切り）の表示／非表示を設定します。属性値「0」を設定すると、フレーム枠は表示されません。「1」を設定すると表示されます。

なお、属性を省略した場合は、自動的に「1」が設定されます。

隣接するフレームのどちらか一方で「FRAMEBORDER="1"」が設定されていると、フレーム枠は表示されます。したがって、フレーム枠を非表示にするためには、両方のFRAME要素で「FRAMEBORDER="0"」を設定する必要があります。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Frameset//EN">
(中略)
<FRAMESET COLS="40%,60%">
    <FRAME SRC="ad_index.html" FRAMEBORDER="0">
    <FRAME SRC="default.html" FRAMEBORDER="0">
</FRAMESET>
```



FRAMEBORDER="0"の例

●MARGINWIDTH属性

フレーム内にHTML文書を表示する際の左右の余白をpixel数で設定します。ただし、表示するフレームのサイズが「設定した左右の余白＋表示されるHTML文書(リソース)」より大きい場合は、見た目は左の余白だけが設定が有効になります(自動的にフレーム枠やウインドウが縮小されることはありません)。

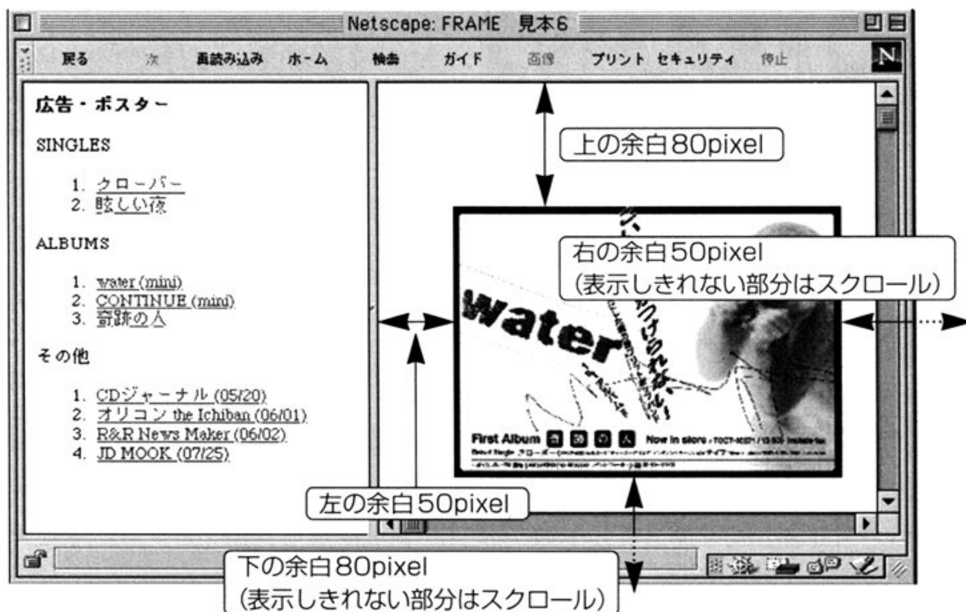
なお、属性を省略した場合の初期値は、Webブラウザによって異なります。

●MARGINHEIGHT属性

フレーム内にHTML文書を表示する際の上下の余白をpixel数で設定します。ただし、表示するフレームのサイズが「設定した上下の余白＋表示されるHTML文書(リソース)」より大きい場合は、見た目は上の余白の設定だけが有効になります(自動的にフレーム枠やウインドウが縮小されることはありません)。

なお、属性を省略した場合の初期値はWebブラウザによって異なります。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Frameset//EN">
(中略)
<FRAMESET COLS="40%,60%">
     <FRAME SRC="ad_index.html">
     <FRAME SRC="jdmook.html" NAME="water_ad"
MARGINWIDTH="50" MARGINHEIGHT="80">
</FRAMESET>
```



余白の設定例

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Frameset//EN">
(中略)
<FRAMESET COLS="40%,60%">
    <FRAME SRC="ad_index.html">
    <FRAME SRC="jdmook.html" NAME="water_ad"
MARGINWIDTH="30" MARGINHEIGHT="20">
</FRAMESET>
```



余白の設定例

拡張

NN 3 IE 3 NN 4 IE 4

NOFRAMES <NOFRAMES>~</NOFRAMES>

フレーム未対応Webブラウザ用の代替要素（The NOFRAMES element）

内容 ▶ BODY（NOFRAMESは除く）

親要素 ▶ FRAMESET

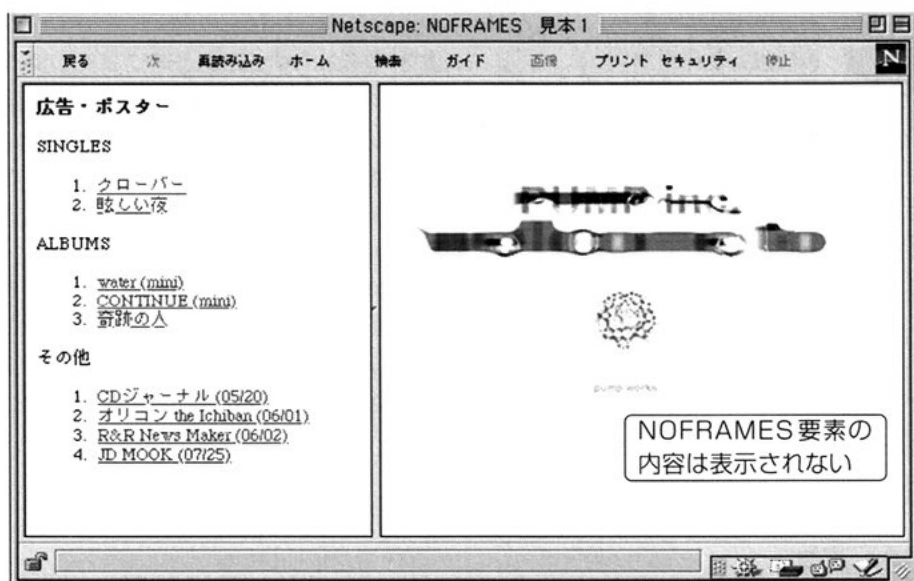
要素の概要

NOFRAMES要素は、フレーム未対応Webブラウザのための代替要素です。開始タグ<NOFRAMES>と終了タグ</NOFRAMES>によってマークアップされた範囲は、通常はフレーム未対応Webブラウザでのみ表示され、フレーム対応Webブラウザでは表示されません。一般的には、別途用意されたフレーム未対応Webブラウザ用のHTML文書へのリンクを設定するために使用します。

NOFRAMES要素は、通常FRAMESET要素の中に置かれます（P.272「注意」参照）。その内容にはBODY要素のみを置くことができますが、BODY要素は開始タグ・終了タグ共に省略が可能です。そのため、見た目上は全てのブロックレベル要素・テキストレベル要素・一般的なテキストを直接置くことができますが、NOFRAMES要素の位置づけを忘れないためにも省略しない方がいいでしょう。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frame
set//EN">
<HTML>
<HEAD>
<TITLE>NOFRAMES　見本１</TITLE>
</HEAD>
<FRAMESET COLS="40%,60%">          ← FRAMESET要素の開始タグ
    <FRAME SRC="ad_index.html">    ← FRAMESET要素
    <FRAME SRC="default.html">
    <NOFRAMES>
    <BODY>                         ← BODY要素の開始タグ
    <P>
    申し訳ありませんが、このページはフレームを使用しております。<BR>
    なお、別途<A HREF="ad_index_nf.html">フレーム不使用のペ
ージ</A>も用意しております。
    </P>
    </BODY>                        ← BODY要素の終了タグ
    </NOFRAMES>
</FRAMESET>                        ← FRAMESET要素の終了タグ
</HTML>
                                   （<P>〜</P>：NOFRAMES要素の内容）
```

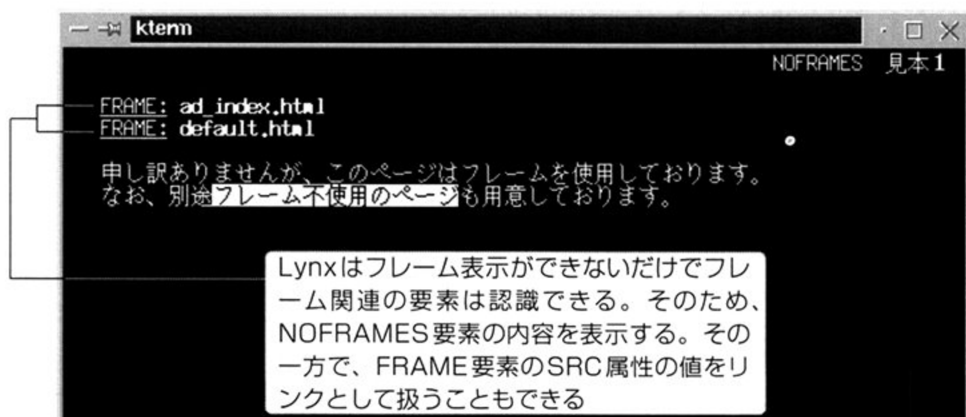
※TITLE要素以外のHEAD要素は省略



フレーム対応のWebブラウザでの表示例



Netscape Navigator 1.1での表示例



Lynxでの表示例

注意 NOFRAMES要素は、フレーム設定用のHTML文書だけでなく、通常のHTML文書（もちろん、HTML 3.2ではなく、HTML 4.0 Transitional DTDに従った文書）の中に置かれることもあります。その場合は、一般的なブロックレベル要素として扱われ、その内容には見出し・アドレス・ブロックレベル要素・テキストレベル要素・一般的なテキストを置くことができます（ただし、NOFRAMES要素の入れ子はできない）。
通常のHTML文書に置かれたNOFRAMES要素の内容は、その文書がFRAME要素に呼び出されてフレーム内で表示される場合には非表示となります。一方、その文書がフレーム未対応のWebブラウザやフレーム機能を使用せずに表示された場合には表示されます。詳細はHTML 4.0の解説書をご覧ください。

属性一覧

省略

属性解説

省略

注意 NOFRAMES要素には、HTML 4.0の基本属性の多くが設定できますが、それらについては別途HTML 4.0の解説書をご覧ください。

NN 3 IE 3 NN 4 IE 4

NOBR

<NOBR>~</NOBR>

改行の無効（NO BReak）　テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DIV CENTER BLOCKQUOTE FORM CAPTION TH TD DT DD LI
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

NOBR要素は、改行を無効にします。たとえば、WebブラウザでHTML文書を表示する場合、通常は自動的に画面の端で折り返されますが、開始タグ<NOBR>と終了タグ</NOBR>タグでマークアップされた範囲は、折り返し表示されなくなります。しかし、Webブラウザのテキストの自動折り返し機能を無効にすると、表示画面の大きさによっては横スクロールしなければならなくなり、不便です。

なお、NOBR要素の中に置かれたBR要素は有効となるようです。

```
<P>
<NOBR>コイツは、手がつけられない！！　全曲シングルかと思わせるクオリティ。他のバンドとはひとまわり違うスケール。<BR>R&amp;B をベースにした実力派ブリットポップバンド water の1st アルバム。</NOBR>
</P>
```



NOBR要素の使用例

属性一覧

なし

属性解説

なし

WBR

<WBR>（終了タグなし）

強制折り返し（WORD BReak） **テキストレベル要素**

内容 空

親要素 BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DIV CENTER BLOCKQUOTE FORM CAPTION
TH TD DT DD LI
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
APPLET FONT

要素の概要

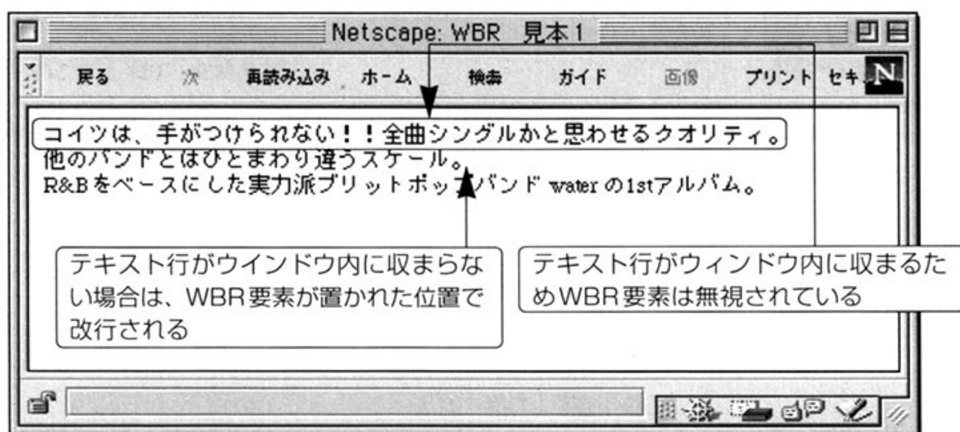
WBR要素は、NOBR要素内で強制的に改行を有効にするための要素です。

この要素は単なる改行機能というより、改行位置の「候補」を設定するような意味合いを持っています。たとえば、BR要素は意識的に改行を行うために使用するものですから、設定された場所で必ず改行されます。しかし、WBR要素の場合はWebブラウザのウインドウに余裕があれば改行されません。また、NOBR要素内でもWBR要素は機能します。

なお、WBR要素は開始タグ<WBR>のみでマークアップされる、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

```
<P>
<NOBR>コイツは、手がつけられない！！<WBR>全曲シングルかと思わせるクオリティ。<BR>
他のバンドとはひとまわり違うスケール。<WBR>R&amp;B をベースにした実力派ブリットポップバンド water の1st アルバム。</NOBR>
</P>
```

WBR要素を指定



WBR要素の使用例

属性一覧

なし

属性解説

なし

MULTICOL <MULTICOL>~</MULTICOL>

コラム（MULTIple COLumns） ブロックレベル要素

内容 TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

親要素 BODY
DIV CENTER BLOCKQUOTE FORM TH TD DD LI

要素の概要

HTML文書でコラム（段組み）を設定する要素です。HTML 3.2で段組みを実現するためには、表を応用するしか方法がありませんが、これは非常に設定が面倒ですし、同時に表が持つ本来の意味とはかけ離れる可能性があります。そこでNetscape社はMULTICOL要素を考案したようですが、独自規格のためほとんど普及していません。

属性一覧

●COLS

値	働き	H3	N4	I4
number	コラム数の設定	−	!	−

属性は必須

●GUTTER

値	働き	H3	N4	I4
pixels	コラム間の余白	−	!	−

属性は省略可

●WIDTH

値	働き	H3	N4	I4
pixels	コラムの横幅	−	!	−

属性は省略可

変数 *number*＝数値　*pixels*＝pixel数

属性解説

●COLS属性

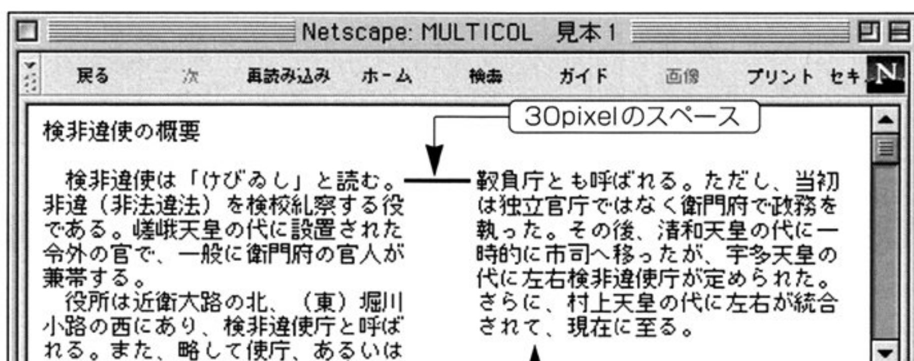
コラム数を設定する必須の属性です。

●GUTTER属性

コラム間の余白をpixel数で設定します。

```
<P>検非違使の概要</P>
<MULTICOL COLS="2" GUTTER="30">
　検非違使は「けびゐし」と読む。非違（非法違法）を検校糺察する役である。
嵯峨天皇の代に設置された令外の官で、一般に衛門府の官人が兼帯する。
<BR>
　役所は近衛大路の北、（東）堀川小路の西にあり、検非違使庁と呼ばれる。
また、略して使庁、あるいは衛門府の別名「靫負のつかさ」から靫負庁とも呼
ばれる。ただし、当初は独立官庁ではなく衛門府で政務を執った。その後、清
和天皇の代に一時的に市司へ移ったが、宇多天皇の代に左右検非違使庁が定め
られた。さらに、村上天皇の代に左右が統合されて、現在に至る。
</MULTICOL>
```

COLS属性、GUITTER属性の指定



GUITTER属性によるコラム幅の指定

●WIDTH属性

コラムの横幅をpixel数で設定します。

WIDTH属性の指定

```
<P>検非違使の概要</P>
<MULTICOL COLS="2" GUTTER="30" WIDTH="250">
  検非違使は「けびゐし」と読む。非違（非法違法）を検校糺察する役である。
(中略)
さらに、村上天皇の代に左右が統合されて、現在に至る。
</MULTICOL>
```



WIDTH属性によるコラム幅の表示例

拡張（独自）

NN 3 NN 4

SPACER

<SPACER>（終了タグなし）

任意のスペース（SPACEing element） テキストレベル要素

内容 ▶ 空

親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P LI DT DD PRE DIV CENTER BLOCKQUOTE
FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
APPLET FONT

要素の概要

HTML文書で任意のスペースを設定する要素です。HTML文書でスペースを実現するためには、よくIMG要素で透過GIF画像を置き、その大きさを適当に調整する方法が使用されています。そのようなトリッキーな方法に比べればSPACER要素は随分スマートですが、やはりNetscape社の独自規格のためほとんど普及していません。

なお、SPACER要素は開始タグ<SPACER>のみでマークアップされる、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

属性一覧

●TYPE

値	働き	H3	N4	I4
horizontal	横方向のスペースを設定（SIZE属性を併用）	−	!	−
vertical	縦方向のスペースを設定（SIZE属性を併用）	−	!	−
block	ブロックスペースを設定 (WIDTH属性とHEIGHT属性を併用)	−	!	−

属性は省略可

●SIZE

値	働き	H3	N4	I4
pixels	テキスト中のスペースの幅	−	!	−

属性は省略可

●WIDTH

値	働き	H3	N4	I4
pixels	ブロックスペースの横幅	−	!	−

属性は省略可

●HEIGHT

値	働き	H3	N4	I4
pixels	ブロックスペースの高さ	−	!	−

属性は省略可

●ALIGN（初期値：bottom）

値	働き	H3	N4	I4
top	テキスト行上端にスペースの上端を揃えて配置	−	!	−
middle	テキスト行下端にスペースの中央を揃えて配置	−	!	−
bottom	テキスト行下端にスペースの下端を揃えて配置	−	!	−
left	テキスト行の左端にスペースを配置し、回り込みを設定	−	!	−
right	テキスト行の右端にスペースを配置し、回り込みを設定	−	!	−

属性は省略可



pixels＝pixel数

属性解説

●TYPE属性

スペースの形状を設定します。

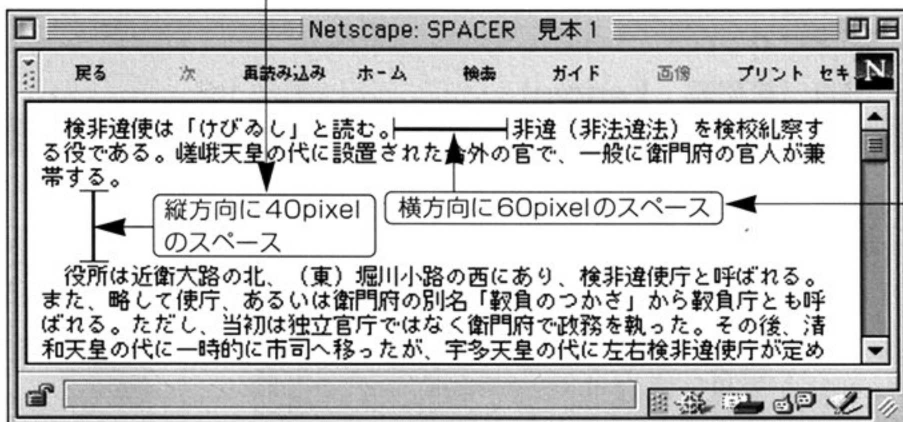
horizontalはテキスト中の横方向のスペース、verticalはテキスト中の縦方向のスペース＝行間設定に相当します。

blockはブロックスペースを設定します。また、ブロックスペースの扱いは、IMG要素で貼り込まれる画像とほぼ同じです。

●SIZE属性

TYPE属性でhorizontalとverticalを設定した場合、SIZE属性で大きさを決定します。

```
<P>
　検非違使は「けびゐし」と読む。<SPACER TYPE="horizontal"
SIZE="60">非違（非法違法）を検校糺察する役である。嵯峨天皇の代に設
置された令外の官で、一般に衛門府の官人が兼帯する。<BR>
<SPACER TYPE="vertical" SIZE="40">
　役所は近衛大路の北、（東）堀川小路の西にあり、検非違使庁と呼ばれる。
（以下略）
```



SPACER要素の使用例

●WIDTH属性

TYPE属性でblockを設定した場合、ブロックスペースの横幅をpixel数で設定します。

●HEIGHT属性

TYPE属性でblockを設定した場合、ブロックスペースの高さをpixel数で設定します。

●ALIGN属性

ブロックスペースとブロックスペースが含まれるテキスト行との位置関係を設定します。ただし、Webブラウザや表示フォントによっては、表現結果が微妙に異なる場合もあります。

なお、ブロックスペースとテキストの位置関係などの表示結果については、IMG要素を参照してください(P.221参照)。

MARQUEE <MARQUEE>~</MARQUEE>

スクロールテキスト (MARQUEE) テキストレベル要素

内容 ▶ TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DIV CENTER BLOCKQUOTE FORM CAPTION
TH TD DT DD LI
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

Microsoft 社の独自規格で、HTML文書内にスクロールテキストを表示するための要素です。マーキーフレーム（枠）内をマーキー（テキスト）が設定された方向へ流れます。

登場した当時は物珍しさからよく使用されていましたが、現在ではほとんど使用されていません。

属性一覧

●WIDTH

値	働き	H3	N4	I4
length	マーキーフレームの横幅	−	−	!

属性は省略可

●HEIGHT

値	働き	H3	N4	I4
length	マーキーフレームの高さ	−	−	!

属性は省略可

●HSPACE

値	働き	H3	N4	I4
pixels	マーキーフレームの横の余白	−	−	!

属性は省略可

●VSPACE

値	働き	H3	N4	I4
pixels	マーキーフレームの縦の余白	−	−	!

属性は省略可

●BGCOLOR

値	働き	H3	N4	I4
color	背景色を指定	−	−	!

属性は省略可

●ALIGN（初期値：bottom）

値	働き	H3	N4	I4
top	テキスト行上端にマーキーフレームの上端を揃えて配置	−	−	!
middle	テキスト行下端にマーキーフレームの中央を揃えて配置	−	−	!
bottom	テキスト行下端にマーキーフレームの下端を揃えて配置	−	−	!

属性は省略可

●DIRECTION（初期値：left）

値	働き	H3	N4	I4
left	マーキーを右から左へ流す	−	−	!
right	マーキーを左から右へ流す	−	−	!
up	マーキーを下から上へ流す	−	−	!
down	マーキーを上から下へ流す	−	−	!

属性は省略可

●BEHAVIOR（初期値：scroll）

値	働き	H3	N4	I4
scroll	マーキーが端から出て端に消える	−	−	!
slide	マーキーが端まで移動したら停止する	−	−	!
alternate	マーキーがマーキーフレーム内を往復する	−	−	!

属性は省略可

●LOOP

値	働き	H3	N4	I4
number	マーキーの表示回数	−	−	!
infinite	マーキーの表示を無限に繰り返す（=-1）	−	−	!

属性は省略可

●SCROLLDELAY

値	働き	H3	N4	I4
number	マーキーの移動時間間隔を ミリ秒数（1/1000秒）で指定	−	−	!

属性は省略可

●SCROLLAMOUNT

値	働き	H3	N4	I4
number	マーキーの移動距離間隔をpixel数で指定	−	−	!

属性は省略可



length＝pixel数、または%　　*pixel*＝pixel数
color＝色指定の16進数、または色名　　*number*＝数値

属性解説

●WIDTH属性

マーキーフレームの横幅をpixel数で、またはWebブラウザのウインドウに対する%値で設定します。

●HEIGHT属性

マーキーフレームの高さをpixel数で、またはWebブラウザのウインドウに対する%値で設定します。

●HSPACE属性

マーキーフレームの横の余白をpixel数で設定します。

●VSPACE属性

マーキーフレームの縦の余白をpixel数で設定します。

●BGCOLOR属性

マーキーフレーム色を16進数、または色名で設定します。

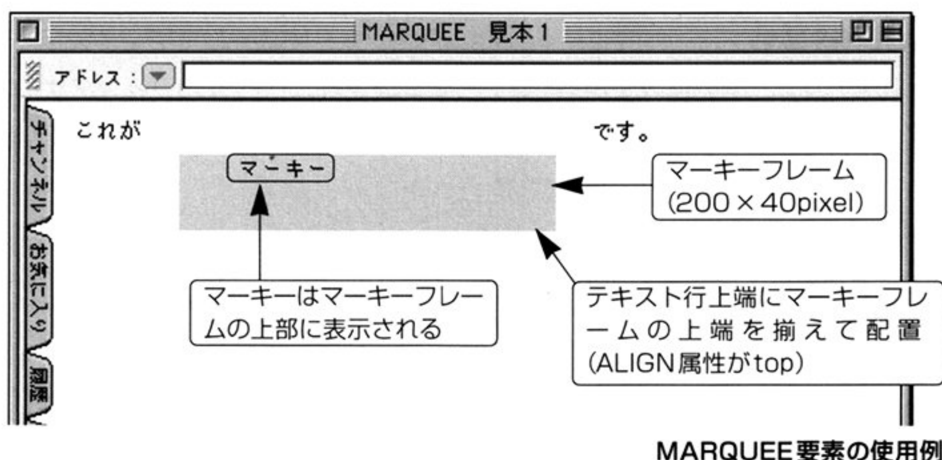
●ALIGN属性

マーキーフレームとマーキーフレームが含まれるテキスト行との位置関係を設定します。ただし、Webブラウザや表示フォントによっては、表現結果が微妙に異なる場合もあります。

属性値top、middle、bottomは、テキスト行に対してマーキーフレームを揃える位置を設定します。また、ALIGN属性を省略すると、初期値としてbottomが使用されます。さらに、HSPACE属性やVSPACE属性でマーキーフレームに余白が設定されている場合は、そこが基準になります。

なお、マーキーフレームとテキストの位置関係などの表示結果については、IMG要素を参照してください(P.221参照)。





MARQUEE要素の使用例

注意 Windows版Internet Explorer 4.0では、ALIGN属性が無視されることがあります。
バージョン3.0、あるいはMac OS版なら問題ありません。

●DIRECTION属性

マーキーを流す方向を設定します。

●BEHAVIOR属性

マーキーの動きに変化をつける属性です。端から出て端に消えるscroll、端まで移動したら停止するslide、マーキーフレーム内を往復するalternateの3種類が設定できます。

●LOOP属性

マーキーの表示回数を設定します。infiniteか-1を設定すると、マーキーの表示を無限に繰り返します。

●SCROLLDELAY属性

マーキーの移動時間、すなわち再描画間隔をミリ秒数(1/1000秒)単位で設定します。

●SCROLLAMOUNT属性

マーキーが一度に移動する距離をpixel数で設定します。値を大きくすると移動速度が上がります。

BLINK

<BLINK>～</BLINK>

テキストの点滅（Blinking text） テキストレベル要素

内容 TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A IMG APPLET FONT BASEFONT BR MAP INPUT SELECT TEXTAREA SCRIPT
一般的なテキスト

親要素 BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER BLOCKQUOTE
FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM STRONG DFN CODE SAMP KBD VAR CITE A APPLET FONT

要素の概要

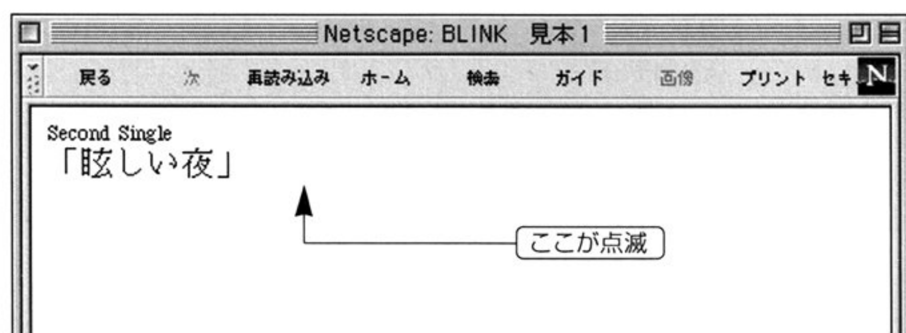
フォントスタイル要素の一種です。

開始タグ<BLINK>と終了タグ</BLINK>でマークアップされたテキストは、画面上で点滅表示されます。古くから存在する要素ですが、あくまでもNetscape社の独自規格です。

テキストを手っ取り早く目立たせるためには便利な要素かもしれませんが、時と場合によっては非常にうるさく感じられます。世の中にはBLINK要素によるテキストの点滅を嫌う人が大勢いることは、知っておいた方がいいでしょう。

```
<P>Second Single<BR>
<FONT SIZE="+2">「眩しい夜」</FONT><BLINK>好評発売中！
</BLINK></P>
```





BLINK要素の表示例

属性一覧

なし

属性解説

なし

EMBED

<EMBED>~</EMBED>

プラグイン（EMBEDding） テキストレベル要素

内容 ▶ NOEMBED
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 ▶ BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER BLOCKQUOTE
FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
APPLET FONT

要素の概要

EMBED要素は、HTML文書内でプラグインを実行させるための要素です。EMBED要素はテキストレベル要素なので、ブロックレベル要素やテキストレベル要素、あるいは一般的なテキストの中に置くことができます。

プラグインが使用するデータや条件は各種属性で設定します。また、開始タグ<EMBED>と終了タグ</EMBED>によって、データの説明やプラグイン未対応のWebブラウザで表示すべき代替テキストをマークアップします（内容をプラグイン対応のWebブラウザで表示させたくない場合は、さらにNOEMBED要素を使用する）。

なお、本書ではEMBED要素の基本的な属性を取り上げています。しかし、プラグインによってはさらに多くの属性を使用して、複雑な表示制御が行えるものもあります。各プラグインについての詳細は、別途プラグイン開発メーカーのマニュアルなどをご覧ください。

属性一覧

●SRC

値	働き	H3	N4	I4
url	プラグインが使用するデータのURL	−	!	!

属性は必須

●WIDTH

値	働き	H3	N4	I4
pixels	表示するデータの横幅	−	!	!

属性は必須

●HEIGHT

値	働き	H3	N4	I4
pixels	表示するデータの高さ	−	!	!

属性は必須

●ALT

値	働き	H3	N4	I4
text	代替テキスト	−	−	!

属性は省略可

●ALIGN（初期値：left）

値	働き	H3	N4	I4
top	テキスト行上端にデータの上端を揃えて配置	−	!	!
middle	テキスト行下端にデータの中央を揃えて配置	−	−	!
bottom	テキスト行下端にデータの下端を揃えて配置	−	!	!
left	テキスト行の左端にデータを配置し、回り込みを設定	−	!	!
right	テキスト行の右端にデータを配置し、回り込みを設定	−	!	!

属性は省略可

●HIDDEN（初期値：false）

値	働き	H3	N4	I4
false	データを表示	−	!	−
true	データを非表示	−	!	−

属性は省略可

●HSPACE

値	働き	H3	N4	I4
pixels	データ表示領域の左右の余白	−	!	!

属性は省略可

●VSPACE

値	働き	H3	N4	I4
pixels	データ表示領域の上下の余白	−	!	!

属性は省略可

●PLUGINSPAGE

値	働き	H3	N4	I4
url	プラグインの入手先のURL	−	!	!

属性は省略可



url＝URL　*pixels*＝pixel数
text＝一般的なテキスト

属性解説

●SRC属性

プラグインが使用するデータのURLを設定する、EMBED要素必須の属性です。

●WIDTH属性

HEIGHT属性と対になって、プラグインの動作領域（データの表示領域＋コントローラ部分）の横幅をpixel数で設定します。属性を省略すると、データの原寸に合わせて自動的に調整して表示されるものもありますが、基本的には設定するものと考えてください。また、プラグインによってはデータの制御用に「コントローラ」を持つものがありますが（QuickTime PluginやMIDI関係のプラグインなど）、その場合は忘れずにコントローラ分のpixel数を加算して設定します。

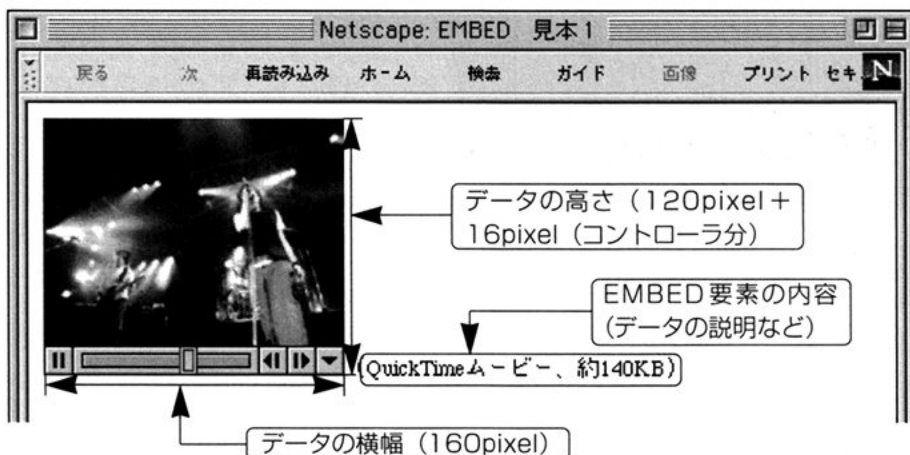
なお、原寸と異なる値を設定することもできますが、その場合は値に合わせて拡大・縮小表示はされません（使用するプラグインによって結果が異なるかもしれません）。

●HEIGHT属性

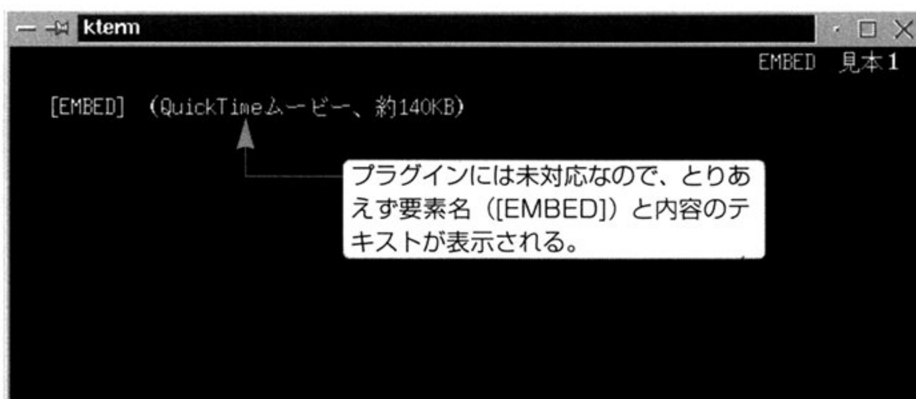
WIDTH属性と対になって、プラグインの動作領域の高さをpixel数で設定します。

```
<EMBED SRC="water_live1.mov" WIDTH="160" HEIGHT="136">
(QuickTimeムービー、約140KB)
</EMBED>
```

データの原寸は160×120pixelだが、QuickTime Pluginを使用する場合は、HEIGHT属性の値にコントローラ分の16pixelを加算する



EMBED要素の使用例

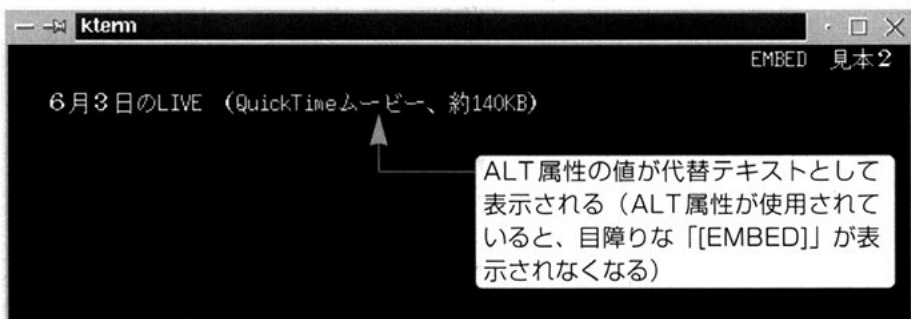


Lynxでの表示例

●ALT属性

主にプラグイン未対応のWebブラウザや音声出力アプリケーションのために、代替テキストを設定する属性です。必須の属性ではありませんが、できるだけ有効に使用してください。

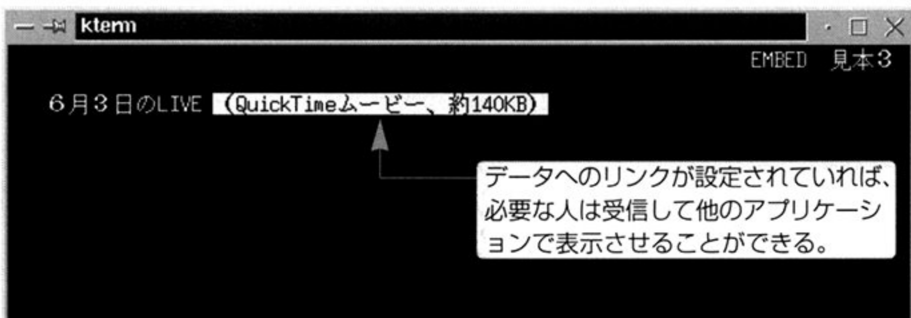
```
<EMBED SRC="water_live1.mov" WIDTH="160" HEIGHT="136"
ALT="6月3日のLIVE">（QuickTimeムービー、約140KB）
</EMBED>
```



ALT属性の使用例

また、プラグイン未対応のWebブラウザでも、データを受信して別のアプリケーションで表示することが可能な場合もあります。そのため、重要なデータやぜひ見てもらいたいデータは、ALT属性の使用と同時に、データへのリンクも設定しておくといいでしょう。

```
<EMBED SRC="water_live1.mov" WIDTH="160" HEIGHT="136"
ALT="6月3日のLIVE">
<A HREF="water_live1.mov">（QuickTimeムービー、約140KB）
</A>
</EMBED>
```



データへのリンクを設定する

●ALIGN属性

プラグインの動作領域とEMBED要素が含まれるテキスト行との位置関係を設定します。ただし、Webブラウザや表示フォントによっては、表現結果が微妙に異なる場合もあります。

属性値top、middle、bottomは、テキスト行に対して画像を揃える位置を設定します。また、ALIGN属性を省略すると、初期値としてbottomが使用されます。

表示結果はIMG要素のALIGN属性を参照してください(P.221参照)。

●HSPACE属性

プラグインの動作領域の左右の余白をpixel数で設定します。属性を省略すると、0以外の最小値が設定されます(具体的な値はWebブラウザによって異なる)。

表示結果は、IMG要素のHSPACE属性を参照してください(P.225参照)。

●VSPACE属性

プラグインの動作領域の上下の余白をpixel数で設定します。属性を省略すると、0以外の最小値が設定されます(具体的な値はWebブラウザによって異なる)。

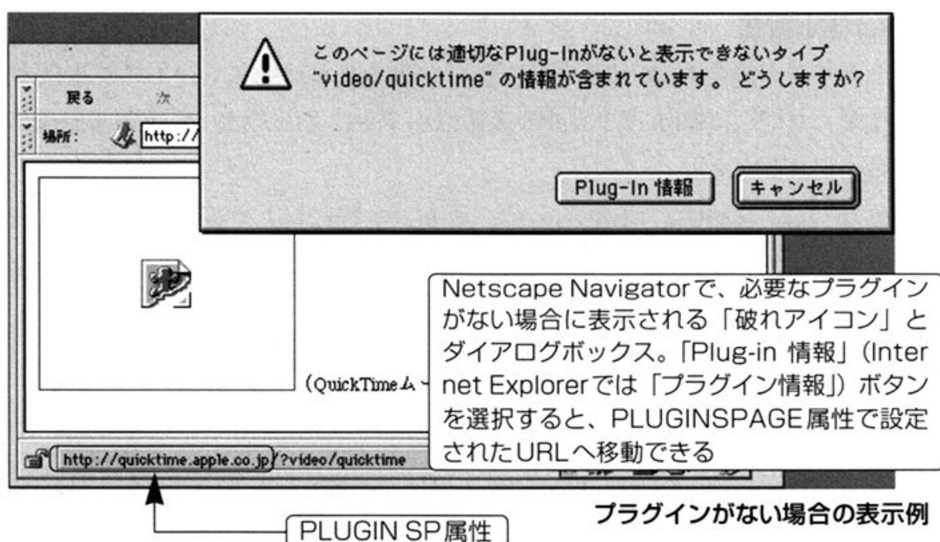
●PLUGINSPAGE属性

プラグインの入手先URLを設定します。

プラグイン対応のWebブラウザで必要なプラグインがインストールされていない場合、プラグインの動作領域に「破れアイコン」が表示されます。また、多くのWebブラウザでは自動的に「必要なプラグインがない」という類のメッセージと共に、プラグインの入手先ページへのリンクボタンが表示されます。

```
<EMBED SRC="water_live1.mov" WIDTH="160" HEIGHT="136"
PLUGINSPAGE="http://quicktime.apple.co.jp/"> (Quick
Time ムービー、約140KB)
</EMBED>
```

※SRC・WIDTH・HEIGHT属性以外の属性は省略



プラグインがない場合の表示例

●HIDDEN属性

属性値にtrueを設定すると、データを非表示にします。属性を省略すると初期値falseが与えられ、データは通常通り表示されます。

NOEMBED <NOEMBED>~</NOEMBED>

プラグイン未対応Webブラウザ用の代替テキスト（NO EMBEDding）　テキストレベル要素

内容 TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE
INPUT SELECT TEXTAREA A IMG APPLET FONT
BASEFONT BR MAP SCRIPT
一般的なテキスト

親要素 EMBED
BODY
H1 H2 H3 H4 H5 H6
ADDRESS
P DT DD LI DIV CENTER BLOCKQUOTE
FORM CAPTION TH TD
TT I B U STRIKE BIG SMALL SUB SUP EM
STRONG DFN CODE SAMP KBD VAR CITE A
APPLET FONT

要素の概要

NOEMBED要素は、EMBED要素の内容をプラグイン対応のWebブラウザで表示したくない場合に使用される要素です。たとえば、プラグイン未対応のWebブラウザのためのメッセージは、対応Webブラウザでは必ずしも必要ではありません。内容によってはかえって紛らわしくなってしまうので、非表示にした方がいいこともあります。

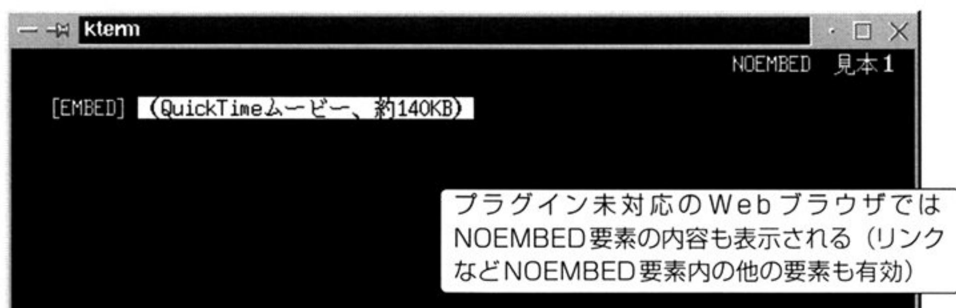
そのような場合にNOEMBED要素が役に立ちます。開始タグ<NOEMBED>と終了タグ</NOEMBED>によってマークアップされた範囲は、プラグイン対応のWebブラウザでは表示されません。もちろん、未対応のWebブラウザでは表示されます。

```
<EMBED SRC="water_live1.mov" WIDTH="160"
HEIGHT="136">
<NOEMBED><A HREF="water_live1.mov">（QuickTimeムービー、
約140KB）</A></NOEMBED>
</EMBED>
```

※SRC・WIDTH・HEIGHT属性以外の属性は省略



プラグイン対応のWebブラウザでの表示例



Lynxでの表示例

属性一覧

なし

属性解説

なし

BGSOUND

<BGSOUND>（終了タグなし）

BGMの利用（BackGround SOUND） ヘッダ要素

内容 空

親要素 HEAD

要素の概要

BGSOUND要素は、サウンドデータをWebページのBGMとして利用するための要素です。

Microsoft社の独自規格のため、Internet Explorerしか対応していません。また、機能的にはEMBED要素でプラグインを利用して同様のことが実現できるため、使用されることはあまりないようです。

なお、BGSOUND要素はヘッダ要素に属しています。これは、サウンドデータを文書の一部に埋め込むというより、「HTML文書全体に対するBGM」という意味合いが強いためでしょう。そのため、EMBED要素と異なりHEAD要素内に置かれます。また、BGSOUND要素は開始タグ<BGSOUND>のみで表現される、いわゆる空の要素です。終了タグは省略ではなく、もともと存在しません。

属性一覧

●SRC

値	働き	H3	N4	I4
url	再生するデータのURL	−	−	!

属性は省略可

●LOOP

値	働き	H3	N4	I4
number	サウンド再生の繰り返し回数	−	−	!
infinite	サウンド再生を無限に繰り返す（=-1）	−	−	!

属性は省略可

変数 *url*＝URL　*number*＝整数

属性解説

●SRC属性

再生するデータのURLを設定します。

●LOOP属性

サウンドの再生を繰り返す回数を設定します。また、infiniteか-1を設定すると、無限に繰り返します。

```
<BGSOUND SRC="clover.wav" LOOP="2">
```

※画面上には何も表示されない

第3章

HTMLの理解のために



3-1 HTMLの簡単な歴史

HTML 2.0

HTMLは、文書の構造を記述(明示)し、内部の情報を正確に伝えたり関連づけする(ハイパーリンク)ためのマークアップ言語として考案されました。そのため文書の表示方法に関しては、あえて規定されていませんでした。

ところが、WWWが一般に普及するに従って、HTML文書を表示したときの「見栄えを制御したい」という要求が強くなってきました。そのため、Webブラウザメーカーでは、自社のWebブラウザでよりきれいに、おもしろくHTML文書を表示するために、さかんにHTMLの拡張を行いました。これは一見ユーザの利益になるように思えます。しかしよく考えると、これはHTML文書の互換性を損なう原因ともなり、「簡単に情報を共有する」というWWWの目的から大きく逸脱することにもなりかねません。

そこで1995年、IETFによって、それまでばらばらに発展していたHTMLの仕様をまとめた共通仕様「HTML 2.0」が策定されました。この規格は現在でも基本的なHTML規格として、多くのWebブラウザに実装されています。

HTML 2.xと3.2

しかし、WebブラウザメーカーはHTML 2.0に満足しませんでした。その後も独自の拡張は続き、再びユーザは便利さと引き換えに混乱を押しつけられてしまったのです。そこで、WWWの標準化と推進のために設けられた国際的な学術研究組織「W3C」において、HTML 2.0の後継規格の審議が開始され、1997年1月には正式に「HTML 3.2」として勧告されました。

HTML 3.2は、当時すでに一般化しつつあったメーカーの独自拡張規格をいくつか採用しています。また、HTML文書を見栄えよく表示したい、という世間の要求にもある程度応えるため、フォントや色の指定方法も実装しています。ただし、見栄えに関しては、やはり本来HTMLが担うべき役割ではありませんし、私たち非西欧人にとっては重大な問題である国際化対応もなされていません

実は、国際化対応については、HTML 3.2とは別にHTML 2.0をベースに進められていました。HTML 2.0では使用できる文字セットがISO 646とISO 8859-1だけで、日本語のJIS X 0208などは埒外に置かれていました。これに対しては、同

じく1997年1月、IETFにおいて正式な国際化規格「HTML 2.x」が追加策定されました。ただし、HTML 2.xは国際化対応しているものの、基本的な機能はHTML 2.0からほとんど変わりがなく、HTML 3.2の代替規格としては世間の要求を満たすことができませんでした。

HTML 4.0

そこでW3CではHTML 3.2の規格から見栄えに関する要素を分離し、加えてHTML 2.xの規格を取り込む形で国際化を果たす「HTML 4.0」規格を提案しました。そして、業界から幅広く意見を求めながら検討を続け、1997年末に正式にW3Cの勧告となりました。

HTML 4.0は真の国際化を果たしているだけでなく、HTML文書の見栄えに関する設定方法の多くをHTMLからスタイルシート（P.307参照）に分離し、HTMLの方は本来の文書の構造記述に専念できるようになっています。また、音声出力の際に混乱のない文書を作成可能にしたり、キーボード操作を容易にする機能の実装など、アクセシビリティが大きく向上した仕様となっています。さらに、スクリプト言語（JavaScriptなど）の正式サポートや、マルチメディアデータの扱いの改善なども大きな特徴となっています。

ところで、一口に「HTML 4.0」といっても、実は3種類のDTD（文書型定義）が用意されています。

●Strict DTD

HTML 4.0の正式な文書型定義です。見栄えの設定に関する要素や属性（ALIGN属性など）をほぼ排除しており、情報共有のためのマークアップ言語としては、理想に近い規格です。

●Transitional DTD

従来のHTML 3.2との互換性を考慮した「過渡的な」規格です。Strict DTDでは、排除された見栄えの設定に関する要素や属性が残されています。そのため、HTML 3.2に従って作成された文書は、比較的容易にTransitional DTDに移行することができます。

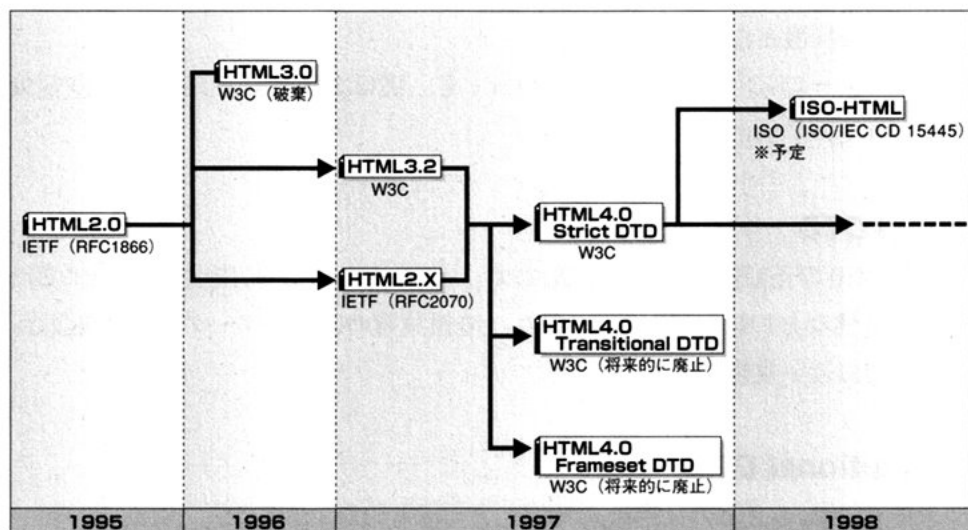
●Frameset DTD

フレームの基本構造を設定する、つまりFRAMESET・FRAME・NOFRAMESの3要素が置かれるHTML文書専用の特別な規格です。作成された

フレーム内に表示されるHTML文書は、Transitional DTDに従ったものになります。

もし、これからHTML 4.0を使用する場合、本当は正式な規格であるStrict DTDに従うことをおすすめしなければなりません。ただし、Strict DTDに従うと、ほとんどの見栄えに関する設定にスタイルシートを併用する必要があります。ところが、現実的にはまだ多くのWebブラウザがHTML 4.0やスタイルシートに十分対応していません。そのため、一般的な利用のためには、現在のところTransitional DTD（必要に応じてFrameset DTD）に従うか、あるいはHTML 3.2や2.0のDTDに従うのが現実的な選択といえます。ただし、Transitional DTDはあくまでも移行措置としての規格で、将来的には廃止される運命にあります。この点は、将来にわたって廃止される予定のないHTML 3.2や2.0と異なります。

また、ISOにおいては、W3Cと協力しながらStrict DTDをベースにした「ISO-HTML」の策定が進められています。ISO-HTMLは、HTML文書の本義に立ち返って、情報の互換性と再利用性を目的とした言語です。そのため、Strict DTDよりさらに厳しく文書構造に合致した記述が求められることになります。なお、ISO-HTMLは1998年末にも「ISO/IEC 15445」として規格化される予定です。



HTMLの変遷

HTML 2.0（RFC 1866　http://ds.internic.net/rfc/rfc1866.txt）
HTML 3.2 Reference Specification（http://www.w3.org/TR/REC-html32）
HTML 4.0 Specification（http://www.w3.org/TR/REC-html40/）
HTML 4.0 Strict DTD（http://www.w3.org/TR/REC-html40/strict.dtd）
HTML 4.0 Transitional DTD（http://www.w3.org/TR/REC-html40/loose.dtd）
HTML 4.0 Frameset DTD（http://www.w3.org/TR/REC-html40/frameset.dtd）

主要なHTMLの仕様書・DTDの入手先

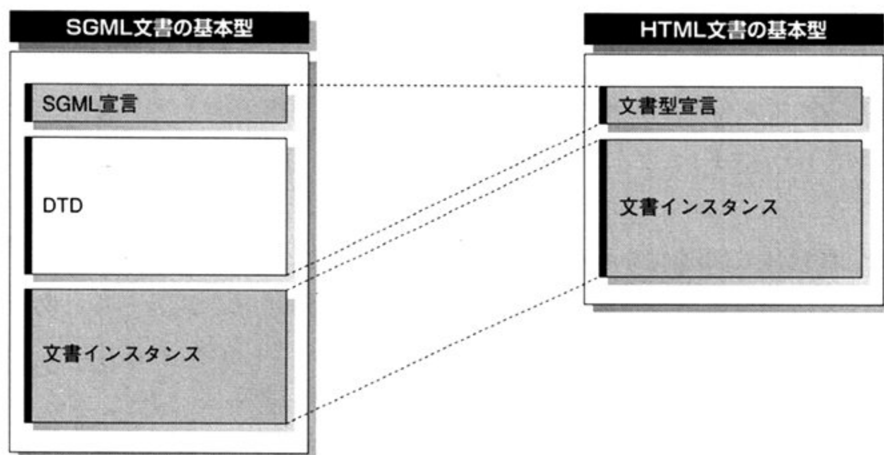
3-2 SGMLとHTMLの関係

実は、HTMLには基になった規格があります。それは、SGMLというマークアップ言語です。SGMLは、文書の共有・互換・再利用のために開発された規格ですから、文書の互換性を損なう最大の原因であるレイアウト情報（表示や印刷など見栄えに関わる情報）を排除し、純粋に文書の構造定義に徹しています。また、文書の作者がマークアップに使用するタグの名前や使用方法をDTDに定義して利用することができるため、文書の互換性を保ちながらも、文書の内容に応じて柔軟に構造定義が行えます。

一方、WWWと共に開発されたマークアップ言語がHTMLです。HTMLは「情報の共有（互換性・再利用性）」というWWWの目的を達成するために、その仕様の基礎部分にはSGMLを採用しました。もっとも、最初のHTMLは要素の定義やタグの使用方法にSGMLの規則を借用した程度で、厳密な規則やDTDは存在しなかったそうです。しかし、HTML 2.0の規格策定にあたっては、厳格にSGMLの規則が適用されました。また、一般ユーザにも使用しやすいように、HTMLではDTDを固定化してしまいました。つまり、言い方を変えれば、DTDが固定化されている面を除いてHTMLは「SGMLそのもの」なのです。

ただ、DTDの固定という仕組みは、一部の人に不満をもたらしました。特に企業からは、固定化されたDTDの枠には収まらない要素に対処すべく「DTDを自由に定義したい」という要望が強く出てきました。そのため、HTMLとは別に新たにXML（eXtensible Markup Language）という規格が、W3Cによって策定され、現在バージョン1.0が正式に勧告されています。とは言え、一般ユーザーがXMLを必要とする時代は、まだ先の話となるでしょう。

なお、本来ならばSGMLやDTDの仕組みを紹介した方がいいのでしょうが、HTML文書の作成に必須の事項ではないことや紙幅の都合もあり、省略させていただきました。興味のある方は、別途SGMLの解説書などをご覧ください。



文書構造面から見たSGMLとHTML

用語解説

・SGML宣言

文書を作成するために使用する基本的な文字セット、文書容量、タグや文字参照に使用される特別な記号に関する規定など基本的な規則が明示されます。たとえば、HTML 3.2では基本的な文字セットがISO 646（ASCII文字セットとほぼ同じ）とISO 8859-1であることなどが宣言されています。

・DTD（Document Type Definition：文書型定義）

どの要素には何というタグをつけてマークアップするのか、要素の親子関係はどうなるのか、属性の値はどうなるのか、などマークアップを行うための基準が記述されています。

SGMLでは、DTDは誰でも作成することができますが、実際には業界で共通のDTDが策定され、それを各企業で利用することも多いそうです。HTMLとは「インターネット業界（IETFやW3C）が作成したDTDをWebブラウザメーカーや一般ユーザが利用している」と考えると、その位置付けがわかりやすいかと思います。

・文書インスタンス（Document Instance）

一般に馴染みのない言葉なので、本書の中では使用していませんが、DTDで定義されたタグなどを使用してマークアップされた文書本体のことをこう呼びます（DTDを具体的に実現したもの）。

・文書型宣言（DOCTYPE宣言）

公的・私的に関わらずDTDが既に存在している場合、その文書インスタンスは一体どのDTDに従って作成されたものかを文書型宣言によって明示し、DTDを省略することができます。

反対に、文書型宣言が欠けていると、その文書インスタンスは何を基準として作成されたのか不明となり、結果的に文書を受け取ってもアプリケーションは解釈ができず、利用することができなくなります。HTMLがSGMLの1形態である以上、文書型宣言が欠かせないものであること理解してください。

3-3 スタイルシートの概要

HTMLには一部を除いて文書の出力様式を厳密に規定する機能はありません。たとえば、「見出しレベル1」という要素をどのように画面表示するかは、Webブラウザの裁量に任されています。ですから、あるWebブラウザでは「見出しレベル1」は大きくて太い文字で表示されますし、別のWebブラウザでは画面の中央に揃えて表示されます。WWWやHTMLが考案された当初はこれでも十分でしたし、現在でもHTML文書の情報を利用するためだけなら本質的には問題はありません。

しかし、WWWが一般に普及するにつれ、HTML文書は情報の共有手段から積極的な宣伝行為にも利用されるようになりました。そうなると、まず大切なのは文書を出力した際の「見栄え」です。いえ、商業行為を伴わなくても、せっかくの情報をよりわかりやすく表現することは大切なことです。そのため、一時期Webブラウザメーカーが HTML文書の「表現力」を高めるために、競って独自のHTMLの拡張規格を作成したこともありました。加えてユーザの側でも、凝ったレイアウトを実現するために、本来のHTMLの規則には合致しない使用方法(裏技)がいろいろと考案されました。

その結果、HTML文書はかなり見栄えのよい表現ができるようになりましたが、その一方で「情報の共有」という本来の目的の維持が危うくなってしまいました。また、Webブラウザによっては実際の見え方が全く異なるため情報が正しく伝わらなかったり、ひどい場合にはまともに表示できなくなる問題も起こってきました。

そこで、導入されたのが「スタイルシート」です。この名前、どこかで見聞きしたことはありませんか？　そう、みなさんが利用しているワープロにもこの機能がありますよね。たとえば「見出し1は新ゴB、18ポイント、斜体、左インデント20mm」とか、「本文はリュウミンM-KL、10ポイント、左インデント25mm」とかあらかじめ登録しておいて、文書を作成する際には半自動的に修飾が要素に適用できる機能です。

HTMLで使用されるスタイルシートも基本的な考え方はこれと同じで、HTML文書内の各要素を個別に、あるいはまとめて修飾するものです。ただ、ワープロのスタイルシートが文書本体と一体不可分の関係にあるのに対して、HTMLで使用するスタイルシートはHTMLとは独立した規格です。そのため、HTML文書の中だけでなく、全く別のファイルに記述して適用することもできる点が異なります。

また、修飾情報の指定は英単語を主体としたテキストファイルで記述されるため、

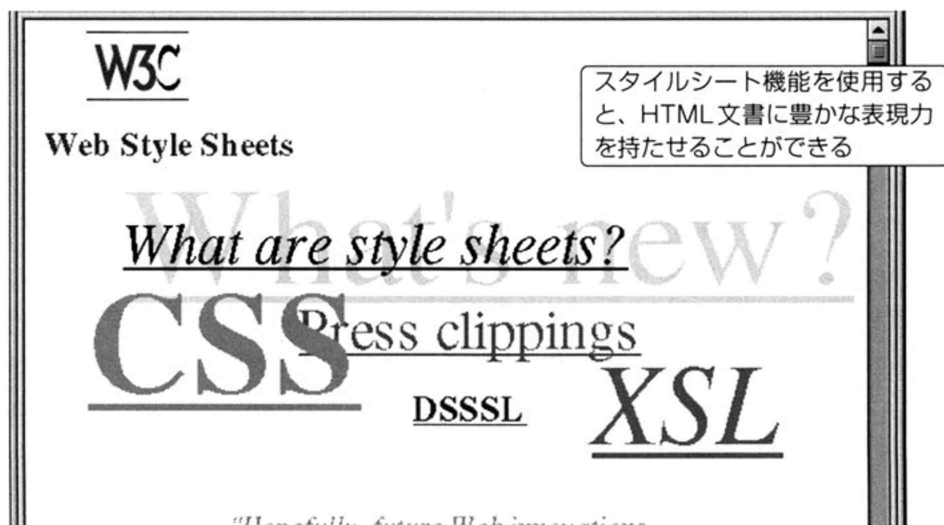
習得や作成が比較的容易、かつスタイルシート自体の互換性や再利用性がある点も異なります。さらにインターネット上で使用されることを考慮して、少しでもデータ量を少なくするための工夫もなされています。

現在のところ、このスタイルシートはCSS1（Cascading Style Sheets, level 1）とその上位規格であるCSS2（Cascading Style Sheets, level 2）の2つがW3Cから正式に勧告されており、HTML 4.0（HTMLの最新バージョン）と組み合わせて使用することができます。

なお、具体的なスタイルシートの使用方法や効果に関しては、別途専門の解説書をご覧ください。



W3Cのスタイルシート解説ページ（スタイルシート機能をオフにして表示）



同ページをスタイルシート機能をオンにして表示

3-4 文字セットと符号化方式

文字セット

現在コンピュータ上では、さまざまな文字が使用されています。もちろん、それらは無秩序に寄せ集めて使用されているのではなく、言語などを単位に使用できる文字の種類が決められています。そのような文字の集まりを「文字セット（※1）」といいます。ローマ字表や常用漢字表、あるいは活字棚のようなものを想像していただければ結構です。

たとえば、米国では「ASCII」という文字セットが標準となっています。ASCII文字セットは、7ビットのデータで表現されますので、2の7乗＝128種類の文字が定義されています（※2）。また、欧州ではデータ量をもう1ビット増やして8ビット（256種類）で表現する「ISO8559-1」という文字セットが使用されています。これは1つの文字セットで欧米の14言語が表現可能なため、インターネット上でも広く使用されています。

日本では、1969年に「JIS X 0201」という文字セットが制定されています。このJIS X 0201には2種類存在します。1つは7ビットのデータを使用するもので、ASCII文字セットとほぼ同じセットと、日本独自のカタカナを収めたセットの2セットを切り替えながら使用するものです。そして、もう1つはデータ量をもう1ビット増やして8ビットにし、ASCII文字セットとカタカナを同じ文字セットに収めたものです。どちらも使用できる合計文字数（256種類）は同じです。なお、一般的に前者を「7ビット系」、後者を「8ビット系」と呼び分けています。

その後コンピュータが進歩してもっと多くの文字数、つまり漢字が扱える可能性が出てきました。しかし、日本語の漢字は8ビットの枠には収まらないので、8ビットのデータを2つ1組で用いて1つの文字を表現する方法が考案されました。これなら理論上は256×256＝65,536種類の文字が表現可能です。もっとも、ASCII文字セットとの互換性を維持するために、実際に定義できる文字数はずっと少なくなります。たとえば、現在の日本で私たちが普段使用している標準的な文字セット「JIS X 0208（※3）」では、8,836種類が定義可能です。

なお、文字セットに関する詳細を知りたい場合は、専門の解説書や資料をご覧ください。とりあえず、ここでは言語などを単位として使用可能な文字種類を定義した「文字セット」というものが存在することを知っていただければ十分です。

符号化方式

文字は文字セットとして定義されただけでは、コンピュータで利用することはできません。文字は具体的なビット配列との対応が定められて、はじめてコンピュータで処理できるようになります。この文字とビット配列を対応させる規則を「符号化方式」といいます。

現在日本で使用されている主な符号化方式には、パソコンでお馴染みの「シフトJIS」、主にUNIX上で使用される「EUC」、そしてインターネットで標準的に使用される「ISO-2022-JP（JISコード）」の3種類があります。また、これらの符号化方式では、JIS X 0201、JIS X 0208など複数の文字セットを同時に扱うことができます。

ところで、インターネット上ではさまざまな言語のHTML文書が公開されています。そのため、サーバやアプリケーションがHTML文書の符号化方式を確実に判断できるように、META要素に符号化方式を記述することが推奨されています（P.40参照）。また、各種符号化方式を明示する場合、charsetパラメータに入れる値（名称）は、以下のようなものがあります。大文字と小文字の区別はありません。

符号化方式	charsetパラメータ値	符号化可能な文字セット
ASCII	us-ascii	ASCII
ISO-8859-1	iso-8859-1	ISO 8859-1
ISO-2022-JP（JISコード）	iso-2022-jp	JIS X 0201　JIS X 0208　他
EUC	euc-jp	JIS X 0201　JIS X 0208　他
シフトJIS	shift_jis	JIS X 0201　JIS X 0208

HTML文書作成ツールの中にはcharsetパラメータ値に「x-euc-jp」や「x-sjis」という値を入れるものがあります。これは「euc-jp」や「shift_jis」が正式名として確定していなかった時代に、Netscape Navigator 2.0が独自に使用した名称の名残りで、困ったことにNetscape Navigator 2.0は正式名称の方を認識しません。ですから、インターネットでHTML文書を公開する場合は、極力ISO-2022-JPだけを使用するのがいいでしょう。

文字参照（特殊記号の取り扱い）

「<」「>」「"」「&」の4種類の記号は、HTMLではマークアップなどに使用されますので（※4）、そのままHTML文書内に記述しても「文字」として認識（表示）されません（たとえば「<」記号はタグの開始記号として解釈される）。では、これらの記号をHTML文書内で「文字」として使用するためには、どうすればいいのでしょ

うか？

HTMLでは、ユーザが使用している符号化方式、あるいはハードウェアやソフトウェアなどの環境によって使用できない文字やHTMLで使用する特殊記号を使用可能にするために「文字参照」という仕組みを用意しています。この文字参照には、文字セットの符号値を使用する「数値文字参照」と、実体名（文字の名前の略称）を使用する「文字実体参照」の２つの方法があります。

もっとも、Webブラウザの表示フォントを「日本語フォント」に設定している場合、文字参照を使用しても現在のWebブラウザでは正しく表示されないことが多いので、あまり使用する機会はありません（※5）。ただ、冒頭の４種類の文字については、日本語フォントを使用していても表示可能なので、覚えておいてください。

数値文字参照では文字コードを、文字実体参照では実体名を「&」と「;」で囲んで参照（指定）します。「;」は条件によっては省略できますが、基本的に常に記述するものと考えてください。また、大文字と小文字は区別されます。

表示文字	文字名	数値実体参照（十進数表記）	文字実体参照
"	二重引用符	"	"
&	アンパサンド	&	&
<	小なり	<	<
>	大なり	>	>

※１）文字セット（Character set）

正式には「文字集合」ですが、一般的には「文字セット」と言った方が通じやすいようなので、本書では文字セットと表記しました。

※２）ASCII
（American national Standard Code for Information Interchage）

US-ASCIIとも呼ばれます。理論上は128種類の文字が使用可能ですが、実際には制御記号への割り当ても必要なので、目に見える文字は94種類です。なお、国際規格としては、通貨記号以外ほぼASCIIそのままのISO 646があります。

※３）

細かく言えば、JIS X 0208にはJIS X 0208-1983とJIS X 0208-1990、JIS X 0208-1997の３種類あり、文字セットの内容が若干異なります。

※４）

属性値の区切り記号「"」は、Webブラウザによってはそのまま表示されることもあります。ただ、混乱を防ぐためも文字参照を使用した方がいいでしょう。

※５）

たとえば、Webブラウザによっては、欧米のWebページを表示すると１バイトカナの「ハ」が表示されることがあると思います。それは「 」という「空白文字」の文字実体参照をWebブラウザが正しく解釈していないことが原因です。

3-5 HTMLの文法チェック

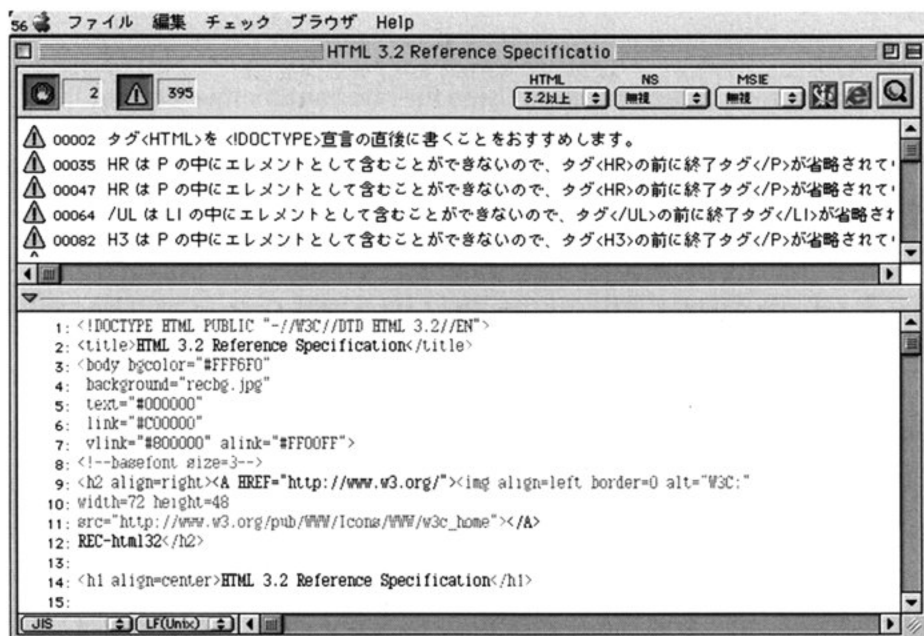
HTML文書が一通りできたら、一般的にはWebブラウザを使用して表示や動作の具合をチェックします。しかし、多くのWebブラウザは一般ユーザ向けの製品だけに、少々HTMLの文法が間違っていてもそれなりに表示してくれます。それはそれで便利なのでしょうが、世の中には正しくマークアップされていないHTML文書からは、正しく情報を引き出せないWebブラウザやアプリケーションもたくさん存在しています。

そこでHTML文書を公開する前に、「HTML文法チェッカー」の類でチェックすることをおすすめします。文法チェックには、アプリケーションを利用したり、文法チェックを行うホームページを利用する方法があります。

●Jchecker

株式会社まつもとが開発したJCheckerは、Mac OS上で動作するチェッカー。HTML 4.0のチェックを主目的にしたチェッカーですが、自分で辞書を調整すればほぼHTML 3.2専用として使用することも可能です。

入手先： **http://www.matsumoto.co.jp/** (2,300円のシェアウェア)



HTML文法チェッカーのJChecker

●Libra（日本語対応HTML文書検証サービス）

矢野啓介氏が作成・公開している検証サービスです。WWWサーバへ送信したHTML文書のURLを入力して、チェックしてもらいます。

```
http://ais2.main.eng.hokudai.ac.jp/%7Eyano/htv/
```

●Another HTML-lint gateway

石野恵一郎氏が作成・公開している検証サービスです。厳密なチェックだけでなく、間違いに対して丁寧な解説もしてもらえるサービスです。また、サーバ上のHTML文書だけでなく、ソースを入力欄にコピー＆ペーストしてチェックしてもらうこともできます。

```
http://ring.aist.go.jp/openlab/k16/htmllint/htmllint.html
http://www.vector.co.jp/htmllint/htmllint.html（ミラーサイト）
```

●i-Checker

日本IBM社が作成・公開している検証サービスです。これはHTMLの文法そのものではなく、主にアクセシビリティに関して重要な項目（たとえば、ALT属性の不備など）についてのチェックを行ってもらえます。

```
http://www.ibm.co.jp/accessibility/i-checker/
```

なお、同社ではアクセシビリティに関する情報を「こころWeb」というサイトで公開しています。実例や関連情報へのリンクも充実していますので、HTML文書を公開する前に、ぜひ一読をおすすめします。

```
http://www.ibm.co.jp/kokoroweb/
```

色名と16進数の対応表

色名	16進数
aliceblue	#F0F8FF
antiquewhite	#FAEBD7
aqua	#00FFFF
aquamarine	#7FFFD4
azure	#F0FFFF
beige	#F5F5DC
bisque	#FFE4C4
black	#000000
blanchedalmond	#FFEBCD
blue	#0000FF
blueviolet	#8A2BE2
brown	#A52A2A
burlywood	#DEB887
cadetblue	#5F9EA0
chartreuse	#7FFF00
chocolate	#D2691E
coral	#FF7F50
cornflowerblue	#6495ED
cornsilk	#FFF8DC
crimson	#DC143C
cyan	#00FFFF
darkblue	#00008B
darkcyan	#008B8B
darkgoldenrod	#B8860B
darkgray	#A9A9A9
darkgreen	#006400
darkkhaki	#BDB76B
darkmagenta	#8B008B
darkolivegreen	#556B2F
darkorange	#FF8C00
darkorchid	#9932CC
darkred	#8B0000
darksalmon	#E9967A
darkseagreen	#8FBC8F
darkslateblue	#483D8B
darkslategray	#2F4F4F
darkturquoise	#00CED1
darkviolet	#9400D3
deeppink	#FF1493

色名	16進数
deepskyblue	#00BFFF
dimgray	#696969
dodgerblue	#1E90FF
firebrick	#B22222
floralwhite	#FFFAF0
forestgreen	#228B22
fuchsia	#FF00FF
gainsboro	#DCDCDC
ghostwhite	#F8F8FF
gold	#FFD700
goldenrod	#DAA520
gray	#808080
green	#008000
greenyellow	#ADFF2F
honeydew	#F0FFF0
hotpink	#FF69B4
indianred	#CD5C5C
indigo	#4B0082
ivory	#FFFFF0
khaki	#F0E68C
lavender	#E6E6FA
lavenderblush	#FFF0F5
lawngreen	#7CFC00
lemonchiffon	#FFFACD
lightblue	#ADD8E6
lightcoral	#F08080
lightcyan	#E0FFFF
lightgoldenrodyellow	#FAFAD2
lightgreen	#90EE90
lightgrey	#D3D3D3
lightpink	#FFB6C1
lightsalmon	#FFA07A
lightseagreen	#20B2AA
lightskyblue	#87CEFA
lightslategray	#778899
lightsteelblue	#B0C4DE
lightyellow	#FFFFE0
lime	#00FF00
limegreen	#32CD32

色名	16進数
linen	#FAF0E6
magenta	#FF00FF
maroon	#B03060
mediumaquamarine	#66CDAA
mediumblue	#0000CD
mediumorchid	#BA55D3
mediumpurple	#9370DB
mediumseagreen	#3CB371
mediumslateblue	#7B68EE
mediumspringgreen	#00FA9A
mediumturquoise	#48D1CC
mediumvioletred	#C71585
midnightblue	#191970
mintcream	#F5FFFA
mistyrose	#FFE4E1
moccasin	#FFE4B5
navajowhite	#FFDEAD
navy	#000080
oldlace	#FDF5E6
olive	#808000
olivedrab	#6B8E23
orange	#FFA500
orangered	#FF4500
orchid	#DA70D6
palegoldenrod	#EEE8AA
palegreen	#98FB98
paleturquoise	#AFEEEE
palevioletred	#DB7093
papayawhip	#FFEFD5
peachpuff	#FFDAB9
peru	#CD853F

色名	16進数
pink	#FFC0CB
plum	#DDA0DD
powderblue	#B0E0E6
purple	#800080
red	#FF0000
rosybrown	#BC8F8F
royalblue	#4169E1
saddlebrown	#8B4513
salmon	#FA8072
sandybrown	#F4A460
seagreen	#2E8B57
seashell	#FFF5EE
sienna	#A0522D
silver	#C0C0C0
skyblue	#87CEEB
slateblue	#6A5ACD
slategray	#708090
snow	#FFFAFA
springgreen	#00FF7F
steelblue	#4682B4
tan	#D2B48C
teal	#008080
thistle	#D8BFD8
tomato	#FF6347
turquoise	#40E0D0
violet	#EE82EE
wheat	#F5DEB3
white	#FFFFFF
whitesmoke	#F5F5F5
yellow	#FFFF00
yellowgreen	#9ACD32

<注意１>　この表は、英語の色名と16進数の相関を知るために使用してください。HTML3.2で定義されている基本16色（表の編掛け部分　P.24参照）以外は、16進数で指定してください。色名で指定した場合は、HTML3.2の文書型宣言は行なえません。

<注意２>　実際に表示（印刷）される色は、利用者側の機器の調整によって大幅に変化します。微妙な色使いにはこだわらないことをお奨めします。

Index

要素引き索引

和文

コメント 29

A

A....206
- HREF 208
- NAME 210
- REL 212
- REV 213
- TARGET....214
- TITLE....213

ADDRESS....54

APPLET 236
- ALIGN 241
- ALT....239
- CODE 238
- CODEBASE....238
- HEIGHT 241
- HSPACE 241
- NAME 240
- VSPACE....241
- WIDTH 240

AREA....230
- ALT....234
- COORDS....231
- HREF 233
- NOHREF 234
- SHAPE....231

B

B....141

BASE....35
- HREF....36
- TARGET 37

BASEFONT 135
- SIZE 135

BGSOUND 299
- LOOP....300
- SRC 300

BIG 147

BLOCKQUOTE....91

BODY 22
- ALINK 25
- BACKGROUND 27
- BGCOLOR 25
- BGPROPERTIES....28
- LEFTMARGIN 28
- LINK 25
- TEXT 25
- TOPMARGIN....28

VLINK 25
BR 60
CLEAR 61
BLINK 288

C

CAPTION 106
ALIGN 108
VALIGN 109
CENTER 90
CITE 170
CODE 162

D

DD 82
DFN 160
DIV 88
ALIGN 89
DL 77
COMPACT 78
DT 81

E

EM 155
EMBED 290
ALIGN 295
ALT 293
HEIGHT 292
HIDDEN 296
HSPACE 295
PLUGINSPAGE 295
SRC 292
VSPACE 295
WIDTH 292

F

FONT 131
COLOR 133
FACE 134
SIZE 132
FORM 172
ACTION 174
ENCTYPE 176
METHOD 175
FRAME 259
FRAMEBORDER 267
LONGDESC 266
MARGINHEIGHT 268
MARGINWIDTH 268
NAME 261
NORESIZE 266
SCROLLING 266
SRC 260
FRAMESET 252
COLS 254
ROWS 253

H

HEAD 21
Hn 51
ALIGN 53
HR 93
ALIGN 94
NOSHADE 94
SIZE 95
WIDTH 95
HTML 20

I

I 139
IMG 215
ALIGN 221
ALT 218
BORDER 224
CONTROLS 227
DYNSRC 226
HEIGHT 223
HSPACE 225
ISMAP 226
LOOP 227
LOOPDELAY 227
SRC 218
START 227
USEMAP 226
VSPACE 226
WIDTH 222
INPUT 179
ALIGN 195
CHECKED 193
MAXLENGTH 194
NAME 188
SIZE 193
SRC 194
TYPE 181
VALUE 190
ISINDEX 33
PROMPT 33

K

KBD 166

L

LI 71
TYPE 74
VALUE 75
LINK 46
HREF 47
REL 47
REV 48
TITLE 49
TYPE 50

M

MAP 228
NAME 229
MARQUEE 283
ALIGN 286
BEHAVIOR 287
BGCOLOR 285
DIRECTION 287
HEIGHT 285
HSPACE 285
LOOP 287
SCROLLAMOUNT 287
SCROLLDELAY 287
VSPACE 285
WIDTH 285
META 40
CONTENT 45
HTTP-EQUIV 41
NAME 44
MULTICOL 277
COLS 278
GUTTER 278
WIDTH 279

N

NOBR 273
NOEMBED 297
NOFRAMES 270
NOSCRIPT 250

O

OL 67
COMPACT 70
START 69
TYPE 68
OPTION 200
SELECTED 200
VALUE 201

P

P 56
ALIGN 57
PARAM 242
NAME 242
VALUE 242
PRE 85
WIDTH 87

S

SAMP 164
SCRIPT（拡張）....................246
CHARSET 249
DEFER 249
LANGUAGE 248
SRC 249
TYPE 247
SCRIPT（ヘッダ情報）....................38
SELECT 196
MULTIPLE 199
NAME 198
SIZE 198
SMALL 149
SPACER 280
ALIGN 282
HEIGHT 282
SIZE 281
TYPE 281
WIDTH 282
STRIKE 145
STRONG 158
STYLE 39
SUB 151
SUP 153

T

TABLE 97
ALIGN 99
BORDER 103
CELLSPACING 104
CELLPADDING 104
BORDERCOLOR 105
BORDERCOLORDARK ..105
BORDERCOLORLIGHT .105
WIDTH 102
TD 122
ALIGN 127
BGCOLOR 130
COLSPAN 126
HEIGHT 129
NOWRAP 124
ROWSPAN 125
VALIGN 128
WIDTH 128

TEXTAREA 203
COLS 205
NAME 205
ROWS 205
WRAP 205
TH 113
ALIGN 118
BGCOLOR 121
COLSPAN 117
HEIGHT 120
NOWRAP 115
ROWSPAN 116
VALIGN 119
WIDTH 119
TITLE 31
TR 110
ALIGN 111
VALIGN 112
TT 137

U

U 143
UL 63
COMPACT 66
TYPE 65

V

VAR 168

W

WBR 275

Index

目的引き索引

文書全体に関わる設定

コメントを記述する 29
自動的に他のリソースを読み込む 44
文書のタイトルを設定する 31
文字の符号化方式を明示する 41
リンク設定の際、基準となるURLを指定する 35

文書本文に関する基本設定

アクセス中のリンク色を指定する 25
既アクセスのリンク色を指定する 25
本文のテキスト色を指定する 24
本文の背景イメージを指定する 27
本文の背景色を指定する 25
未アクセスのリンク色を指定する 25

文書の基本構造

Webブラウザの自動テキスト折り返し機能を無効にする 273
引用文を区別する 91
改行位置を指定する 60
改行位置の候補を指定する 275
空白などで整形したテキストファイルをそのままの形で表示する 85
作者の連絡先（アドレス）を記述する 54
水平線を引く 93

段組みを指定する	277
段落を作成する	56
任意のスペースを設定する	280
複数の要素をまとめて位置揃えする	88
見出しを指定する	51

リスト

語句とその説明をリスト形式で表示する	77
番号付きリストの開始番号を指定する	69,75
番号付きリストの行頭記号を変える	68,74
番号付きリストを表示する	67
番号なしのリストの行頭記号を変える	65,74
番号なしのリストを表示する	63

表

表題を指定する	106
表のサイズを指定する	102
表のセルサイズを指定する	119,128
表のセル内での改行を禁止する	115,124
表のセル内のデータの位置を指定する	111,118,127
表の外枠の太さを指定する	103
表のデータを設定する	122
表の表示位置を指定する	99
表の見出しを設定する	113
表の枠組みを指定する	97
表の枠の太さを指定する	104
複数の行にまたがるセルを作成する	116,125
複数の列にまたがるセルを作成する	117,126

フォーム

1行のテキスト入力欄を設定する	182
選択メニュー形式のフォームで複数項目の選択を許可する	198

選択メニュー形式のフォームを設定する	196
単数項目を選択するボタン(ラジオボタン)を設定する	183
データフィールドに名前を設定する	189
デフォルトで選択されるボタン項目を指定する	193
デフォルトで選択されるメニュー項目を指定する	200
入力内容をクリアするボタンを設定する	186
入力内容を送信するボタンを設定する	184
パスワード入力タイプのフィールドを設定する	182
フォームによるデータの送信方法を指定する	176
複数行のテキスト入力欄を設定する	203
複数項目を選択するボタン(チェックボックス)を設定する	183

フォントとフォントスタイル

上付添字を指定する	153
基準となるフォントサイズを指定する	135
下付添字を指定する	151
スクロールテキストを表示する	283
テキストを点滅表示する	288
等幅フォントで表示する	137
フォント色を指定する	133
フォントにアンダーラインを付加する	143
フォントに取消線を付加する	145
フォントのサイズを指定する	132
フォントの種類を指定する	134
フォントをイタリック表示する	139
フォントを大きめに表示する	147
フォントを小さめに表示する	149
フォントをボールド表示する	141

語句への意味付け

引用・参照元文書のタイトルなどを明示する	170
語句を強調する	155
語句を強く強調する	158

サンプル出力であることを明示する 164
定義された語句であることを明示する 160
プログラムコードであることを明示する 162
プログラムの変数であることを明示する 168
ユーザ(キーボード)が入力すべき語句であることを明示する 166

リンク

HTML文書内にリンク先を設定する 210
テキストや画像にリンクを設定する 208
複数のHTML文書のリンク関係を設定する 212
リンク機能を使用してメールを送信する 209

画像関連

画像に対するテキストの回り込みを解除する 61
画像に対するテキストの回り込みを指定する 221
画像の代わりに表示するテキストを設定する 218
画像の表示サイズを指定する 222
画像を表示する際の余白を指定する 225
テキスト行内の画像の表示位置を設定する 221
表示する画像を指定する 215

クライアントサイド・イメージマップ

クライアントサイド・イメージマップ用の画像を指定する 229
クライアントサイド・イメージマップを設定する 230

Java Applet

Java Appletを使用する 236

スクリプト

使用するスクリプト言語を設定する 246

スクリプト言語未対応Webブラウザへの配慮 250

フレーム

新たにウィンドウを開いてリンク先のHTML文書を表示する 264
フレーム境界線の表示を設定する 267
フレーム内に表示するデータを設定する 260
フレーム内のスクロールバーを制御する 266
フレームの基本構造を設定する 253
フレーム未対応ブラウザのユーザーへ配慮する 270
フレームを解除してリンク先のHTML文書を表示する 264
ユーザによるフレームサイズの変更を禁止する 266
リンク先のリソースを表示するターゲットフレーム名を指定する 261
リンク先を表示する基準フレーム名を指定する 263

プラグイン

プラグイン未対応Webブラウザへの配慮する 297
プラグインを使用してデータを表示する 290

HTMLハンドブック　改訂第2版

1998年10月1日　第1刷発行

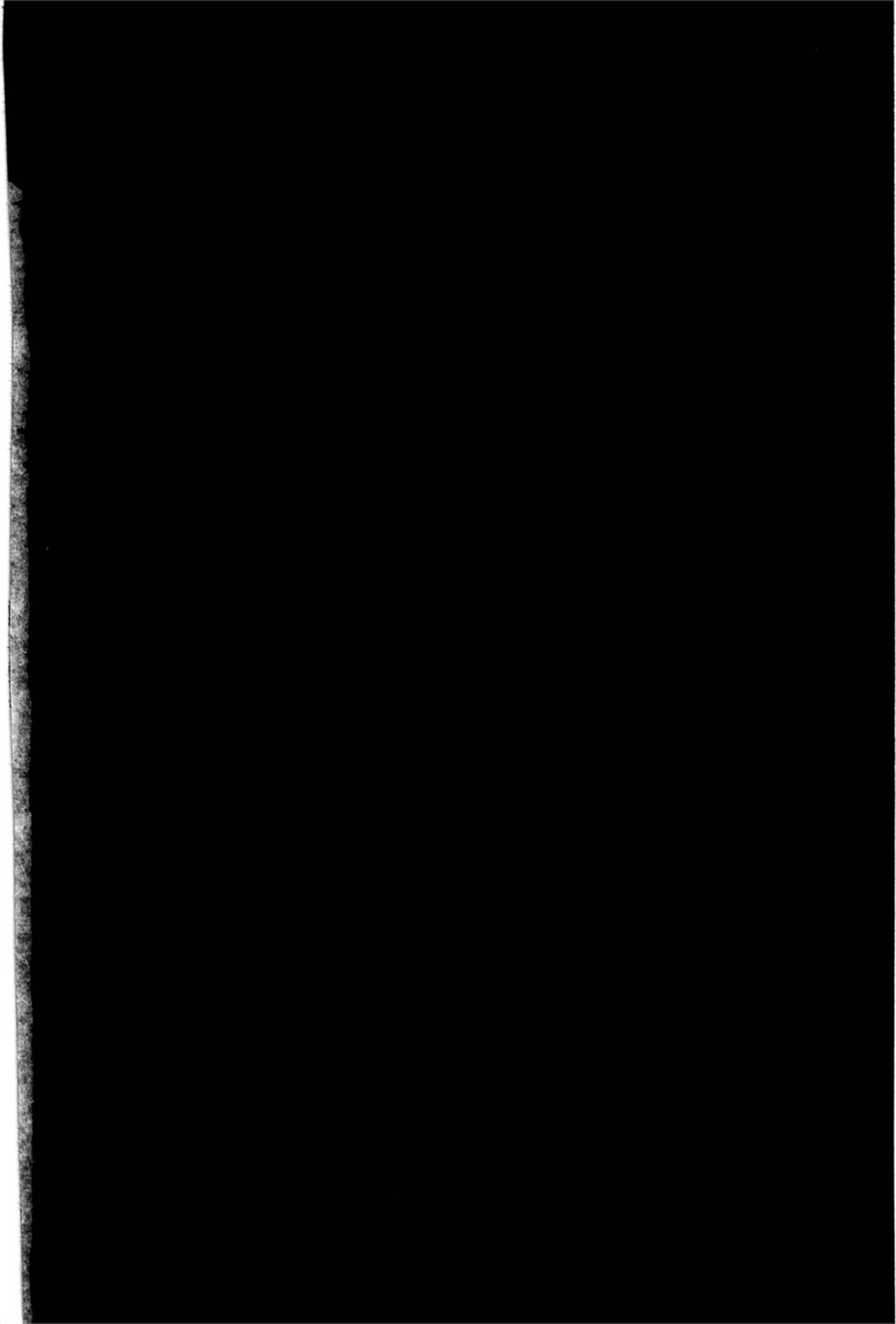
著　者 ……………… 渡辺　竜生（わたなべ　たつお）
発行者 ……………… 岡崎　眞
発行所 ……………… ソフトバンク株式会社 出版事業部
〒103-8501　東京都中央区日本橋箱崎町24-1
電話　販売局　03（5642）8101
編集局　03（5642）8148
組　版 ……………… Creaters Cable（横山浩一）
印　刷 ………………クニメディア株式会社

カバーデザイン …… 尾崎　真人
カバーイラスト …… Mitsuhiko Yamamoto
本文イラスト ……… 佐藤　元

執筆協力　小沢正直・後藤昌代・嶋岡正二郎
water（江沼　学・宮永治郎・鎌田雅人・岡本慎吾・吉田太郎）

落丁本、乱丁本は小社販売局にてお取り替えいたします。
定価はカバーに記載されております。
本書内容に関するご質問などは、ご面倒でも小社第2編集部まで、必ず書面にてご連絡くださいますようお願いいたします。

Printed in Japan　ISBN 4-7973-0692-0



即戦力ハンディガイド



一太郎8
ハンドブック
浅賀幸一著



Excel97
ハンドブック
西沢夢路著



Word97
ハンドブック
鈴木良子著



Access97
ハンドブック
辻　一著



Windows95
ハンドブック
阿部信行著



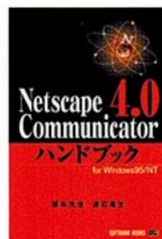
PowerPoint97
ハンドブック
MbCD著



Outlook97
ハンドブック
木地本昌弥著



Excel97関数
ハンドブック
西沢夢路著



Communicator4.0
ハンドブック Win版
坂本光世 渡辺竜生著



InternetExplorer4.0
ハンドブック
山口　学著



Windows95
ハンドブック IE4対応版
尾崎行雄著



Word98
ハンドブック
鈴木良子著



Outlook98
ハンドブック
木地本昌弥著



Windows98
ハンドブック
西沢夢路著

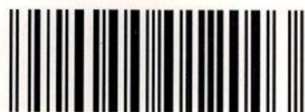


Windows98ハンドブック
ネットワーク&ハードウェア編
阿部信行 尾崎行雄著

SOFT BANK Publishing ソフトバンク パブリッシング

定価 本体1,200円 +税





ISBN4-7973-0692-0

C0055 ¥1200E

HTML HANDBOOK

